

Uso de geolocalización y de fuentes de datos abiertas para la creación de servicios turísticos por la ciudad de Madrid

Juan Bautista Arellano Bruno

Máster en Internet De Las Cosas
Septiembre de 2019 con una calificación de 8.5



Trabajo de Fin de Máster

Director: Antonio Sarasa Cabezuelo

Autorización de difusión

Juan Bautista Arellano Bruno

Fecha

El abajo firmante, matriculado en el Máster en de Internet De Las Cosas de la Facultad de Informática, autoriza a la Universidad Complutense de Madrid (UCM) a difundir y utilizar con fines académicos, no comerciales y mencionando expresamente a su autor el presente Trabajo Fin de Máster: “Uso de geolocalización y de fuentes de datos abiertas para la creación de servicios turísticos por la ciudad de Madrid”, realizado durante el curso académico 2018-2019 bajo la dirección de Antonio Sarasa Cabezuelo en el Departamento de Sistemas Informáticos y Computación, y a la Biblioteca de la UCM a depositarlo en el Archivo Institucional E-Prints Complutense con el objeto de incrementar la difusión, uso e impacto del trabajo en internet y garantizar su preservación y acceso a largo plazo.

Resumen

Este proyecto constituye el Trabajo de Fin de Máster de Juan Bautista Arellano Bruno, alumno del Máster de Internet De Las Cosas de la Universidad Complutense de Madrid.

El objetivo del proyecto es aprovechar la geolocalización de los móviles y las fuentes de datos abiertas para crear servicios de valor añadido orientadas hacia las visitas turísticas en la ciudad de Madrid. En este sentido, las fuentes de datos abiertos se usan para recuperar elementos de interés cultural en la ciudad de Madrid tales como museos, monumentos o parques.

Como principal resultado del proyecto, se ha desarrollado una aplicación Android capaz de recuperar información sobre elementos de interés cultural de la ciudad de Madrid y mostrar aquellos elementos que se encuentran dentro de una determinada distancia desde la posición en la que se encuentra en un momento dado el usuario.

Así mismo, la aplicación hace uso de la API de la EMT de Madrid para poder buscar rutas, bien sea andando o usando el transporte público, hasta los elementos de interés cultural desde la ubicación del usuario.

Palabras clave

Fuentes de datos abiertos.

Geolocalización.

Android.

API.

Abstract

This project constitutes the Master's Final Project by Juan Bautista Arellano Bruno, student of the Internet Master's Degree in Internet Of Things at the Complutense University of Madrid.

The objective of the project is to take advantage of the geolocation of mobile phones and open data sources to create value-added services oriented towards tourist visits in the city of Madrid. In this sense, open data sources are used to recover elements of cultural interest in the city of Madrid such as museums, monuments or parks.

As the main result of the project, an Android application has been developed capable of recovering information on elements of cultural interest of the city of Madrid and showing those elements that are within a certain distance from the position in which it is at a given time the user.

Likewise, the application makes use of the EMT API of Madrid to be able to search for routes, either walking or using public transport, to the elements of cultural interest from the user's location.

Key words

Open data sources

Geolocation

Android

API

Table of Contents

<i>Uso de geolocalización y de fuentes de datos abiertas para la creación de servicios turísticos por la ciudad de Madrid</i>	<i>1</i>
<i>Autorización de difusión</i>	<i>2</i>
<i>Resumen</i>	<i>3</i>
Palabras clave	3
<i>Abstract</i>	<i>4</i>
Key words	4
<i>Capítulo 1</i>	<i>7</i>
1.1 Introducción	7
1.2 Motivación	7
1.3 Objetivos	8
1.4 Plan de trabajo	8
<i>Chapter 1.....</i>	<i>10</i>
1.1 Introduction	10
1.2 Approach	10
1.3 Objectives	11
1.4 Working Plan	11
<i>Capítulo 2 – Estado del arte</i>	<i>13</i>
2.1 Sensores y Smartphone	13
2.2 Fuentes de datos abiertos	15
2.2.1. Datos abiertos	15
2.2.2. La Iniciativa de datos abiertos del Gobierno de España.....	17
2.2.3. El portal de datos abiertos del ayuntamiento de Madrid	17
2.2.4. El repositorio de datos abiertos de la EMT	18
2.3 Geolocalización.....	18
2.4 Proyectos similares.	20
2.4.1 Madrid’s Best: City Travel Guide	20
2.4.2 Mapa de Madrid offline	20
2.4.3 Google Maps	21
2.4.4 Foursquare	21
2.4.5 Yelp.....	21
<i>Capítulo 3. Desarrollo del proyecto.....</i>	<i>23</i>
3.1 Marco de trabajo	23
3.2 Especificación de requisitos	23
3.2.1 Actores del sistema	23
3.2.2 Casos de uso	24
3.3 Arquitectura	34
3.4 Modelo de datos.....	35
3.4.1 Modelo Entidad-Relación	35

3.4.2 Base de datos	35
Capítulo 4. Diseño de la aplicación.....	38
4.1 Registro de usuarios	38
4.2 Búsqueda de elementos de interés cultural cercanos	40
4.3 Búsqueda de elementos de interés cultural de Madrid	43
Capítulo 5. Herramientas tecnológicas	44
5.1 Lenguajes de programación	44
5.1.1 Python.....	44
5.1.2 Java	44
5.1.3 PHP.....	45
5.2 Librerías utilizadas	45
5.2.1 PHPMailer	45
5.2.2 Volley	45
5.3 Entornos de desarrollo.....	46
5.3.1 Android Studio	46
5.3.2 Google Cloud Platform.....	47
5.4 Formatos de datos.	48
5.4.1 JSON	48
5.4.2 XML	48
5.5 Herramientas auxiliares	48
5.5.1 REM.....	48
5.5.2 Visual Paradigm	48
5.5.3 Sublime Text 3.....	49
Capítulo 6. Conclusiones y trabajo futuro.....	50
Chapter 6. Conclusions.....	52
Bibliografía	53
Apéndice 1 - Manual de usuario	57
A1.1 Introducción	57
A1.2 Pantalla de inicio	57
A1.3 Pantalla de registro	58
A1.4 Pantalla de recuperación de contraseña	59
A1.5 Pantalla de inicio	60
A1.6 Pantalla del perfil del usuario	64
A1.7 Pantalla de favoritos.....	65
A1.8 Pantalla de ajustes	65

Capítulo 1

1.1 Introducción

La idea básica del concepto de IOT es la utilización de objetos cotidianos tales como frigoríficos, ropa o televisiones que están conectados a la red y con los que se puede interactuar con el objetivo de aprovechar la información generada por los mismos. Se busca desarrollar soluciones en las que la comunicación es máquina a máquina, y que por tanto, no sea necesaria la interacción humana para realizar las tareas para las que están pensadas [\[1\]](#).

La tecnología IOT se está aplicando tanto en el entorno empresarial como en el entorno cotidiano. Por ejemplo, en el entorno empresarial se utiliza en plantas de producción donde diferentes dispositivos y sensores conectados, permiten recoger datos generados en dicha planta, para posteriormente ser analizados, y así generar alarmas o mensajes para que se tomen las acciones correspondientes en función del tipo de alarma o mensaje generado. Una aplicación más cotidiana podría ser la conexión de un electrodoméstico, por ejemplo, el frigorífico, a Internet con el fin de avisar sobre qué alimentos van a caducar próximamente, o incluso, avisar en caso de que se produzca una bajada de temperatura por alguna avería.

Los Smartphone son un ejemplo de dispositivos que se puede relacionar con el IOT. Los Smartphone actuales poseen numerosos sensores capaces de recoger información valiosa del entorno. Gracias a la información que se recoge por medio de estos sensores, se pueden crear servicios de valor añadido [\[2\]](#).

Este proyecto busca aprovechar los sensores de un Smartphone para poder crear un servicio de valor de añadido que permita al usuario disponer de información acerca de elementos de interés cultural que se encuentran cercanos a una localización determinada, y poder optimizar la visita de estos.

1.2 Motivación

La forma de viajar y realizar visitas culturales ha evolucionado según ha ido evolucionando la tecnología.

Tradicionalmente cuando se llegaba a una ciudad nueva, se buscaba en primer lugar una oficina de turismo en la que se preguntaba a un experto sobre cuáles eran los elementos de interés cultural más importantes de la zona. Esta persona, sobre un mapa en papel, los rodeaba y daba consejos sobre la forma óptima de cómo llegar hasta ellos y cómo visitarlos, obligando así a los turistas a tener que orientarse por una zona desconocida con un mapa. En caso de que se quisiese conocer de forma detallada la información de alguno de ellos, era necesario, o bien leer una guía turística en papel en la que se detallaba extensamente la información relativa a dicho elemento, o una vez se llegaba hasta este, participar en una visita organizada (en ocasiones pagando) donde un experto le daba a conocer toda la información.

Con la aparición de los Smartphone y el acceso a Internet, este método se ha visto modificado. Acudir a una oficina de turismo se ha convertido en una opción pues gracias a un Smartphone y al acceso a Internet, se puede de una forma rápida y cómoda conocer cuáles son los elementos de interés cultural de una ciudad y cómo llegar a ellos. Así mismo, es posible disponer de valoraciones y opiniones de otros turistas que han visitado previamente esa zona (hace años a esta información sólo se podía acceder si se conocía directamente a una persona que hubiese estado en dicha zona). De esta forma, los Smartphone han supuesto una gran mejora en el turismo, pues se puede conocer información y ver un elemento de interés cultural sin la necesidad de tener que acudir a él y en función de los gustos del turista, acudir a visitarlo personalmente o no.

En este proyecto se propone aprovechar la información a la cual se tiene acceso desde un móvil para crear un servicio de valor añadido que utilice la información de geolocalización que ofrece un Smartphone con el objetivo de mostrar la ubicación e información de elementos de interés cultural que se encuentran cerca del usuario.

1.3 Objetivos

El principal objetivo del trabajo consiste en realizar una aplicación móvil que aprovechará la geolocalización que facilita un móvil, la información sobre museos y monumentos de la ciudad de Madrid que se encuentran en el portal de datos abiertos del Ayuntamiento de Madrid y la fuente de datos abiertos de la EMT de Madrid, para crear un servicio de valor añadido orientado a facilitar las visitas culturales. Este objetivo se puede matizar en los siguientes objetivos más concretos:

1. Facilitar la búsqueda de museos o monumentos de la ciudad de Madrid cercanos a la localización de un usuario.
2. Mostrar de una forma visual información (ubicación, información de acceso, comercios cercanos...) acerca de los museos o monumentos seleccionados por un usuario.
3. Buscar la ruta de acceso desde el punto en el que se encuentra el usuario a un museo o monumento seleccionado utilizando la red de autobuses de la ciudad de Madrid.
4. Permitir que los usuarios puedan valorar los monumentos o museos marcándolos como favoritos.

1.4 Plan de trabajo

Para realizar el proyecto se ha seguido el siguiente plan de trabajo:

1. Búsqueda de información sobre el estado del arte. Durante esta fase se recopiló información sobre los distintos sensores que son frecuentes en los Smartphone así

como de las aplicaciones más comunes de estos. Además, se buscó información sobre las fuentes de datos abiertos y cuáles eran las que mejor se adaptaban a las necesidades del proyecto, y, por último, se realizó una investigación sobre aplicaciones que propusiesen soluciones similares a la planteada con este proyecto.

2. Especificación de requisitos. En esta fase se fijó la funcionalidad que se iba a implementar el proyecto.
3. Búsqueda de fuentes de datos abiertos. En esta fase se analizaron las fuentes de datos abiertas que posee el portal de datos abiertos del Ayuntamiento de Madrid con el objetivo de valorar qué información sobre elementos de interés cultural de Madrid poseen. En particular, se puso especial atención en museos, monumentos, parques, iglesias y oficinas de turismo.
4. Diseño y desarrollo de una aplicación para Android. En esta fase se llevó a cabo la implementación de la aplicación móvil de acuerdo a la especificación establecida.
5. Evaluación técnica de la aplicación. En esta fase, se ha sometido a diferentes pruebas la aplicación con el objetivo de solucionar problemas que pudieran surgir y comprobar el correcto funcionamiento de la misma.

En la Figura 1 se puede apreciar el cronograma que muestra la planificación de las actividades que se han llevado a cabo, así como las fechas en las que se estima que se acabe cada una de ellas.

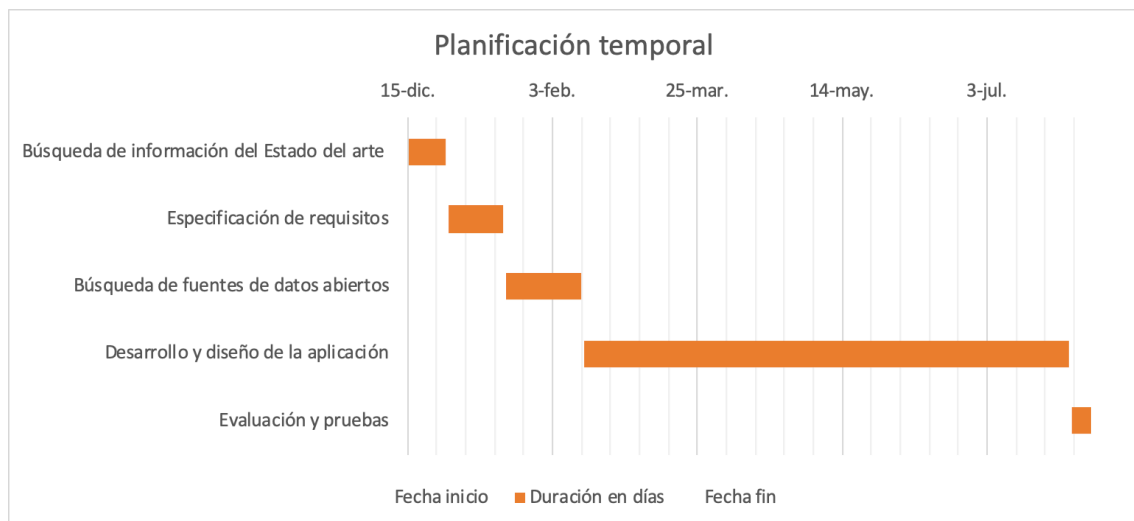


Figura 1 Planificación temporal

Chapter 1

1.1 Introduction

The basic idea of IOT concept is the use of everyday objects such as refrigerators, clothes or televisions that are connected to the network and with which you can interact with the objective of taking advantage of the information generated by them. It seeks to develop solutions in which communication is machine to machine, and therefore, human interaction is not necessary to perform the tasks for which they are intended [1].

IOT technology is being applied both in the business environment and in the everyday environment. For example, in the business environment it is used in production plants where different devices and connected sensors, allow to collect data generated in said plant, to later be analyzed, and thus generate alarms or messages so that the corresponding actions are taken depending on the type of alarm or generated message. A more everyday application could be the connection of an appliance, for example, the refrigerator, to the Internet in order to warn about what foods are going to expire soon, or even notify in case of a temperature drop due to a breakdown.

Smartphone is an example of devices that can be related to the IOT. T Current Smartphone has numerous sensors capable of collecting valuable information from the environment. Thanks to the information collected through these sensors, value-added services can be created [2].

This project seeks to take advantage of the sensors of a Smartphone to be able to create an added value service that allows the user to have information about elements of cultural interest that are close to a specific location, and to optimize their visit.

1.2 Approach

The way of travel and cultural visits has evolved as technology has evolved.

Traditionally when a new city was reached, a tourist office was first sought in which an expert was asked about where the most important elements of cultural interest in the area were. This person, on a paper map, surrounded them and gave advice on the optimal way to reach them and how to visit them, thus forcing tourists to have to orient themselves through an unknown area with a map. In case you wanted to know in detail the information of any of them, it was necessary, or read a tourist guide on paper that detailed the information related to that element, or once it was reached, participate in an organized visit (sometimes paying) where an expert gave him all the information.

With the appearance of the Smartphone and Internet access, this method has been modified. Going to a tourist office has become an option because thanks to them, you can quickly and conveniently know where the elements of cultural interest of a city are and how to reach them. Likewise, it is possible to have ratings and opinions of other tourists who have previously visited that area (years ago this information could only be

accessed if a person who had been in that area was known directly). In this way, the Smartphone has meant a great improvement in tourism, because you can know information and see an element of cultural interest without the need to go to it and depending on the tastes of the tourist, go to visit it personally or do not.

In this project it is proposed to take advantage of the information that is accessed from a mobile phone to create an added value service that uses the geolocation information offered by a Smartphone with the aim of showing the location and information of elements of cultural interest that are close to the user.

1.3 Objectives

The main objective of the project is to make a mobile application that will take advantage of the geolocation provided by a mobile, information about museums and monuments of the city of Madrid that are in the open data portal of the City of Madrid and the source of open data from the EMT of Madrid, to create an added value service aimed at facilitating cultural visits. This objective can be qualified in the following more concrete objectives:

1. Facilitate the search for museums or monuments in the city of Madrid near the location of a user.
2. Visual display information (location, access information, nearby shops ...) about the museums or monuments selected by a user.
3. Search the path from the point where the user is to a selected museum or monument using the bus network of the city of Madrid.
4. Allow users to rate monuments or museums by marking them as favorites.

1.4 Working Plan

To carry out the project, the following work plan has been followed:

1. Search for information on the state of the art. During this phase, information was collected on the different sensors that are frequent in the Smartphone as well as the most common applications of these. In addition, information was sought on open data sources and which ones were best suited to the needs of the project, and, finally, an investigation was conducted on applications that proposed solutions similar to that proposed with this project.
2. Requirements specification. In this phase the functionality that was going to implement the project was fixed.

3. Search for open data sources. In this phase, the open data sources of the open data portal of the Madrid City Council were analyzed in order to assess what information on elements of cultural interest in Madrid they have. In particular, special attention was paid to museums, monuments, parks, churches and tourist offices.

4. Design and development of an Android application. In this phase the implementation of the mobile application was carried out according to the established specification.

5. Technical evaluation of the application. In this phase, the application has been subjected to different tests in order to solve problems that may arise and check the correct operation of the application.

Figure 1 shows the schedule that shows the planning of the activities that have been carried out, as well as the dates on which it is estimated that each one of them is finished.

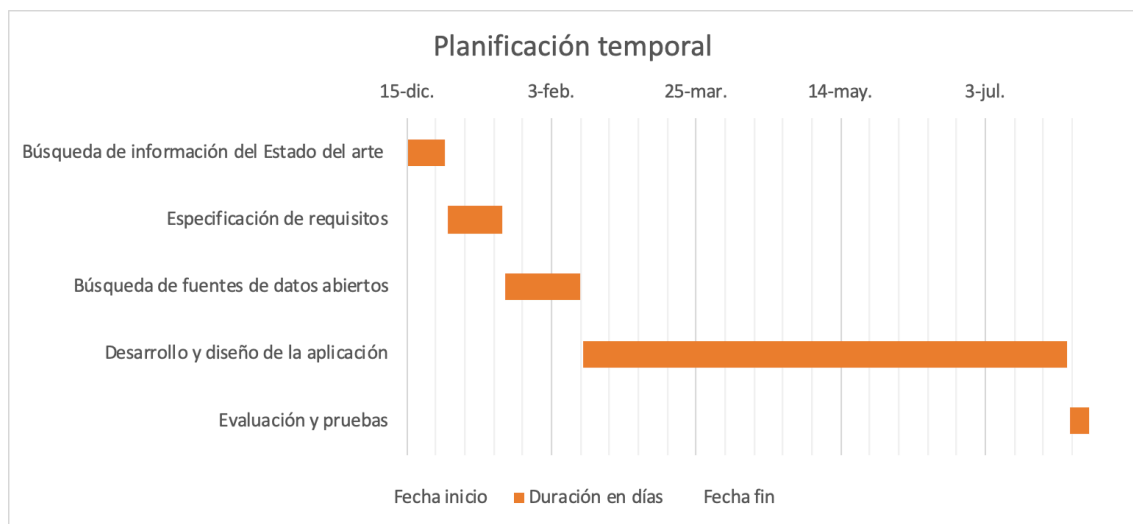


Figure 1 Working Plan

Capítulo 2 – Estado del arte

En este capítulo se hará una revisión del estado del arte relacionado con el desarrollo del proyecto. En particular, se comentarán los sensores más comunes que se encuentran en los Smartphone y sus principales aplicaciones en el ámbito de la IOT, se explicarán en qué consisten los repositorios de datos abiertos y su importancia, se describirá en qué consiste la geolocalización de dispositivos y su aplicación en los Smartphones, y por último se mostrarán aplicaciones que realizan tareas similares a la aplicación desarrollada en este proyecto.

2.1 Sensores y Smartphone

Un sensor es un dispositivo capaz de captar información de una magnitud del exterior, y transformarla en otra magnitud, por lo general eléctrica, que se pueda manipular y trabajar con ella [3]. En cuanto a la información que se puede recuperar del entorno, se puede mencionar la temperatura o humedad del ambiente, así como la ubicación y la orientación en la que se encuentra un determinado dispositivo.

A lo largo de la evolución que han sufrido los Smartphone, se les ha ido dotando cada vez de más sensores capaces de recopilar una gran cantidad de información del entorno. Los sensores que poseen estos dependen del fabricante. A continuación, se van a explicar un conjunto de sensores que son comunes a la mayoría de los teléfonos.

El sensor que posee todo teléfono ya sea Smartphone o no, es el micrófono. Un micrófono es un sensor capaz de transformar el sonido que este percibe en una señal eléctrica. Es el sensor que permite captar la voz en una llamada telefónica, y junto con el altavoz, el cual, realiza la tarea opuesta que el micrófono, poder establecer conversaciones telefónicas. Una aplicación de este tipo de sensor dentro del concepto IOT puede ser la de captar el ruido ambiente que hay en una determinada zona, por ejemplo, cerca de un hospital, para asegurarse de que no se sobrepasan los decibelios permitidos y que no se moleste a los pacientes de dicho hospital. Otra aplicación de este tipo de sensor puede ser la captura del sonido ambiente para poder determinar qué canción está sonando.

Un sensor también importante con el que cuentan la mayor parte de los Smartphone es el acelerómetro. Este sensor es capaz de poder determinar la orientación en la que se encuentra colocado un dispositivo, con lo que se consigue saber si el dispositivo está colocado de manera horizontal, vertical o incluso, si este está colocado boca abajo[4]. Otra de las aplicaciones que puede tener este tipo de sensores es en la industria cinematográfica. Gracias a este tipo de sensor, se puede disponer de cámaras, hasta cierto, inmunes al pulso humano, consiguiendo así un video más estabilizado y con mayor calidad.

Otro sensor muy ligado al anterior es el giroscopio. Este tipo de sensor permite obtener información sobre la aceleración no gravitacional, y que está destinado a complementar la información recogida por el acelerómetro. Este tipo de sensor añade una cuarta

dimensión al movimiento, de tal manera, que se permite conocer el giro y rotación del Smartphone [4]. En cuanto a las aplicaciones que puede tener este tipo de sensor, decir que gracias a la información que recoge, un Smartphone con giroscopio puede actuar como ratón de un ordenador [5].

El sensor de proximidad es un sensor con el que cuentan la mayor parte de los Smartphone. Este sensor está formado por LED infrarrojo que emite un rayo no perceptible por el ser humano y un receptor de infrarrojos capaz de detectar la vuelta del rayo cuando este rebota con una superficie. Calculando el tiempo que tarda el rayo en rebotar, se puede determinar si existe un objeto cerca de este. En los Smartphone se usa para determinar si se está hablando por teléfono para apagar la pantalla y poder ahorrar así batería [4]. Una aplicación más relacionada con IOT, sería la de aplicar este tipo de sensores dentro de garajes de aparcamiento. Un sensor de proximidad colocado en cada una de las plazas podría detectar si hay un coche aparcado, o si, por el contrario, está libre, y conectadas con un sistema de luces, encender una luz verde o roja en función del estado de la plaza y que el conductor tardase menos tiempo en aparcar el vehículo.

De igual modo, el sensor de luminosidad está en gran cantidad de dispositivos. Este sensor mide la cantidad de luz en el entorno en unidades con las que se pueda trabajar [6]. Gracias a este tipo de sensor, el brillo de la pantalla de los Smartphone se puede ajustar a las condiciones lumínicas del entorno, evitando así que los usuarios puedan sufrir daño en los ojos. Este tipo de sensor es de igual modo muy utilizado en domótica, donde se usa para poder realizar tareas en función de la cantidad de luz que hay, para así, por ejemplo, poder abrir las persianas de forma automática por la mañana para amanecer con la luz del día, y cerrarlas al anochecer para poder tener privacidad en el hogar y evitar ser vistos desde fuera.

Otro tipo de sensores que están teniendo una gran importancia son los sensores biométricos, los cuales están dedicados a identificar las características físicas e incluso comportamiento de una persona. La principal función que buscan cumplir estos sensores es la de aportar seguridad a los sistemas y evitar que personas ajenas a estos hagan uso de ellos. Suponen una gran revolución en cuanto a tradicionales métodos de seguridad, ya sean los números PIN, tarjetas magnéticas y contraseñas [7]. Este tipo de sensores incluyen numerosos sensores de diversos tipos:

1. Lector de huellas. Este sensor permite crear una reproducción de la huella dactilar, y una vez almacenada, compararla con la reproducción de la huella de los intentos de acceso para verificar si la persona que quiere acceder al sistema es la correcta [6].
2. Lector de Iris. Gracias a una luz infrarroja imperceptible para el ojo humano, este tipo de sensor permite analizar los patrones del iris del ojo humano, y poder así compararlos con los que se tienen almacenados en el sistema para comprobar si la persona que desea acceder al sistema puede acceder de verdad [6].
3. Sistemas de desbloqueo facial. Estos sistemas combinan un conjunto de cámaras de alta calidad para analizar los patrones faciales del rostro del usuario. De igual modo a

los anteriores, tras analizarlos, compara con los patrones almacenados para poder determinar si el usuario puede acceder o no al sistema [6].

Por último, el sensor GPS del móvil. Este sensor es capaz de geolocalizar el dispositivo gracias a la señal que emiten los satélites que orbitan alrededor de la Tierra. Al leer estas señales se busca triangular la posición [4]. Por regla general, para obtener la localización de una manera precisa, este sensor utiliza los ángulos de inserción de la señal de al menos tres satélites diferentes. Las funcionalidades que ofrece este sensor son opcionales, es decir, en caso de que el usuario no desee estar posicionado, este sensor se puede desactivar.

2.2 Fuentes de datos abiertos

2.2.1. Datos abiertos

Los datos abiertos son conjuntos de datos producidos o recopilados por diferentes organismos que se ponen a disposición de la ciudadanía para que se puedan utilizar libremente de manera sencilla. Este tipo de datos desempeñan un importante papel a la hora de proporcionar transparencia a organismos públicos, así como de proporcionar igualdad en oportunidades a la hora de crear riqueza [8]. Se pretende que los datos abiertos sean datos que cualquiera puede usar y distribuir siempre y cuando se respeten el reconocimiento de su autoría [9].

Los datos abiertos deben de cumplir una serie de principios, sin los cuales, no se podrían considerar abiertos. Algunos de estos principios son los siguientes:

1. Los datos abiertos son relevantes solo si estos están actualizados y son comprensibles.
2. Deben de poder ser accesibles, es decir, los datos deben de poder ser encontrados de una manera sencilla, gratuita y distribuirse bajo una licencia abierta.
3. Comparables. Los datos abiertos deben de poderse contrastar y comparar con otros. Este principio permite aumentar el valor de estos al favorecer la veracidad de estos con otras fuentes de información.
4. Los datos abiertos permiten mejorar la confianza que tiene la ciudadanía sobre los gobiernos al poder comprobar que tareas realizan estos.
5. Para el desarrollo e innovación. Los datos abiertos permiten el desarrollo de nuevos servicios y aplicaciones.

Una de las propiedades más importantes de los datos abiertos es la interoperabilidad. La interoperabilidad otorga la propiedad de que diversos sistemas y organizaciones puedan

trabajar juntos. Esta habilidad es esencial para poder desarrollar sistemas complejos, pues sin ella, es bastante complicado conseguirlo [\[10\]](#).

Los repositorios de datos abiertos están pensados para servir a personas individuales y a empresas para que puedan usar la información pública. En concreto, se puede usar para:

1. Simples consultas.
2. Enriquecer la información con nuevos datos, es decir, crear valor añadido.
3. Desarrollar aplicaciones y servicios.
4. Crear nuevos negocios.

A pesar de las numerosas ventajas que presentan los datos abiertos, estos en ocasiones pueden presentar problemas [\[11\]](#). Los problemas más frecuentes que se pueden dar a la hora de trabajar con datos abiertos son:

1. Licencias no abiertas o poco claras. En numerosas ocasiones, los datos tienen asociados una licencia restrictiva o desconocida, impidiendo de esta manera, la reutilización de estos.
2. Conjuntos de datos incompletos. La mayoría de los datos se encuentran divididos entre múltiples secciones de los sitios en los que estos se publican.
3. Formatos difícilmente reutilizables. El uso de formatos estandarizados es limitado, lo que hace más difícil la interoperabilidad.
4. Datos desactualizados. En ocasiones los datos no se actualizan con la frecuencia que la naturaleza de los datos indica.
5. Tasas de acceso. Es habitual tener que pagar para poder disponer acceso total a los datos publicados, lo que favorece la brecha digital.

En lo que se refiere a los datos abiertos, existen tipos diferentes tipos basándose en el ámbito en el que se puedan aplicar. Entre los tipos más comunes, se pueden mencionar:

1. Cultural. Hace referencia a datos sobre obras y bienes culturales.
2. Ciencia. Hace referencia a datos que se producen de la investigación científica.
3. Finanzas. Incluyen datos como las cuentas de gobierno, y la información sobre datos financieros.
4. Estadísticas. Datos generados por las oficinas de estadística.
5. Tiempo. Datos que se usan para predecir el clima.

6. Medio ambiente. Datos relativos al medio ambiente.

7. Transporte. Incluyen datos sobre horarios o rutas.

En concreto, para este proyecto interesan los repositorios de datos abiertos de tipo cultural para recuperar información sobre los elementos de interés cultural, y de transporte, para recuperar información de rutas sobre como llegar a dichos elementos de interés cultural usando el transporte público.

2.2.2. La Iniciativa de datos abiertos del Gobierno de España

A nivel nacional, uno de los repositorios de datos abiertos más importantes que se puede encontrar es el de la Iniciativa de datos abiertos del Gobierno de España (<https://datos.gob.es>), o como también se le conoce, Iniciativa Aporta. Este repositorio es el encargado de organizar y gestionar el catálogo de información del sector público [12]. Esta iniciativa tiene como objetivo promover la cultura de la publicación de datos abiertos de las distintas administraciones, las cuales, son las encargadas de publicar los datos [13].

En cuanto al tipo de datos que ofrece, decir que ofrece múltiples tipos de datos, entre los que se puede mencionar, entre otros:

1. Datos del sector público.
2. Medio ambiente.
3. Demografía, sociedad y bienestar.
4. Ciencia y tecnología.
5. Vivienda.
6. Economía.
7. Turismo y comercio

2.2.3. El portal de datos abiertos del ayuntamiento de Madrid

El portal de datos abiertos del ayuntamiento de Madrid (<https://datos.madrid.es>) está dedicado a promover el acceso a los datos del gobierno municipal e impulsar el desarrollo de herramientas creativas para servir a la ciudadanía de Madrid. Este repositorio no está pensado para un usuario final como pueden ser otros portales, para los que ya se disponen otras herramientas como puede la web del ayuntamiento de Madrid (www.madrid.es), sino que está orientado a aquellas personas o entidades que quieren hacer algo distinto con los datos de Madrid [14].

Este portal proporciona información sobre una extensa gama de edificios, eventos, instalaciones o actividades que se pueden encontrar en la ciudad de Madrid, es por ello por lo que este repositorio se ha usado para recuperar información sobre los elementos

de interés cultural de Madrid, entre los que cabe mencionar, museos, monumentos, parques, oficinas de turismo e Iglesias.

2.2.4. El repositorio de datos abiertos de la EMT

Este repositorio proporciona una gran cantidad de información en cuanto al transporte público en la ciudad de Madrid permitiendo así poder utilizar información de este tipo en el desarrollo de aplicaciones y sistemas propios. En concreto, es el operador de movilidad de Madrid que se encarga de la gestión del autobús urbano de la ciudad [47]. El repositorio, en el que se debe registrar previamente para poder utilizar, permite conocer información sobre las paradas de una ruta de autobús, horarios de los autobuses de cada línea o incluso, incidentes que se produzcan o el tiempo de espera estimado para que un autobús llegue a una determinada parada.

2.3 Geolocalización

La geolocalización es la capacidad de obtener la ubicación real de un objeto en el ámbito geográfico de manera muy precisa. La geolocalización está relacionada con los sistemas de posicionamiento, con la diferencia que esta se centra más en poder determinar una posición significativa, como puede ser la dirección de una calle, en vez de enfocarse solo en conocer las coordenadas geográficas [15].

Hay que remontarse hasta los años 30 para encontrar los primeros sistemas de localización. Estos se basaban en un receptor de radio con una antena que determinaba la dirección de la señal de llegada y calculaba el retraso con el transmisor. Pero no fue años más tarde, en concreto hasta el uso de satélites, cuando esta tecnología dio un salto importante en cuanto a precisión. En la actualidad conviven principalmente el sistema de posicionamiento global de EE. UU. (GPS) y Navegación por satélite de Rusia (GLONASS). Además, existe un programa estratégico mundial llamado Sistema de navegación global por satélite (GNSS) [16].

En cuanto a las técnicas que se pueden emplear para conocer la ubicación de un objeto, decir que son tres principalmente [17].

En primer lugar, la localización GPS. Es el tipo de localización que se consigue gracias a la red de más de 30 satélites que orbitan alrededor de la Tierra y que en conjunto, tienen una visión global de la Tierra. El método con el que se consigue obtener la ubicación del dispositivo es la siguiente. Cada satélite emite una señal de forma periódica marcando la localización real del dispositivo, y teniendo en cuenta altitud, longitud, altura y tiempo, se obtiene la ubicación. Cuantos más satélites formen parte en el proceso, más precisa será la ubicación. El GPS que posee el teléfono móvil, o los navegadores GPS hacen uso de este tipo de localización. Este, es un sistema de geolocalización muy preciso capaz de ubicar una dirección de forma exacta.

Otro método es la localización GSM. Este tipo de localización tiene lugar gracias a las torres y antenas de telefonía sin las cuales, los móviles no tendrían cobertura. Estas antenas permiten rastrear un teléfono para geolocalizarlo, aunque de manera poco precisa. Con este método se puede marcar la zona en la que el dispositivo se encuentra, pero no se puede conocer la ubicación exacta. Según el caso, el margen de error puede ser de hasta 200 metros.

Por último, otro método por el que se puede conocer la ubicación de un dispositivo es a través de la geolocalización WIFI. La geolocalización GPS funciona muy bien al aire libre, pero en interiores, es decir, dentro de edificios, presenta problemas en lo que se refiere a la precisión de la ubicación. La geolocalización wifi permite solucionar este problema. El dispositivo manda una señal a las redes cercanas, aunque estas estén cerradas, estas señales viajan por el aire a una velocidad conocida, por tanto, el tiempo entre una transmisión que sale de un punto de acceso hasta llegar al dispositivo se puede calcular. Combinando la información de cada una de las redes, se puede, por tanto, conocer la ubicación del dispositivo [18].

Gracias al avance de los métodos de localización, se han conseguido desarrollar aplicaciones y servicios capaces de explotar esta información tanto en favor del usuario, como a favor empresarial.

Dentro del sector empresarial, alguno de los fines con los que se puede explotar la localización de un dispositivo pueden ser [16]:

1. Geotargeting. Consiste en la búsqueda de posibles clientes potenciales de un determinado negocio en función de su ubicación geográfica con el fin de proporcionar un contenido segmentado.
2. Beaconing. Permite que empresas puedan enviar notificaciones a través de Bluetooth cuando usuarios pasan cerca de un determinado establecimiento.
3. Localización de pedidos. Numerosas empresas emplean la geolocalización para determinar la ubicación en el momento que se desee de un determinado paquete, aspecto.

Lo que el sector empresarial busca con estos fines de explotación de la localización es fidelizar o captar clientes. La localización se convierte en un activo muy importante para las empresas. Es por ello, que existen numerosas aplicaciones que explotan la localización que proporciona el sensor GPS de los Smartphone.

Un tipo de aplicaciones que hacen un uso intensivo de la localización son las redes sociales. Aplicaciones como, por ejemplo, Instagram o Facebook, a la hora de subir contenido a estas plataformas, hacen uso de la función de geolocalización para permitir etiquetar con la ubicación dicho contenido, de tal manera que, si un usuario realiza una búsqueda de contenido, poder ver el contenido tomado en esa ubicación. Twitter es otro ejemplo de red social que hace uso de la geolocalización ya que cada tweet, puede llevar

asociada la ubicación desde la que este fue publicado. Por último, WhatsApp, permite compartir la ubicación en tiempo real con otro usuario.

Lo que se pretende al mezclar geolocalización y redes sociales es lo que se conoce como geo-socialización [19].

La industria de los videojuegos también es capaz de explotar la localización en sus productos. Pokemon Go [20] es un videojuego, desarrollado por Niantic, de realidad aumentada que se basa en la localización. Este videojuego consiste en buscar y capturar Pokemons de la saga, que se encuentran escondidos en ubicaciones del mundo real, lo cual, implica tenerse que desplazar distintas zonas para poder avanzar.

Por último, otro tipo de aplicaciones que hacen uso este tipo de información son las aplicaciones que permiten encontrar tus dispositivos perdidos. Este tipo de aplicaciones permiten dar a conocer la ubicación en la que se encuentra el dispositivo en caso de pérdida o robo de este, incluso sin tener el sensor GPS del Smartphone activado.

2.4 Proyectos similares.

Durante un periodo de investigación, se han encontrado aplicaciones Android que realizan tareas similares a las que se realizan con este proyecto. A continuación, se describen.

2.4.1 Madrid's Best: City Travel Guide

Madrid's Best: City Travel Guide [21] es una aplicación de Android que posee un mapa interactivo con una serie de marcadores que representan los elementos de interés cultural de la ciudad Madrid. El usuario puede interactuar con dichos marcadores para obtener información detallada de los elementos de interés cultural, horarios, descripciones, precios...

Adicionalmente, decir que se presenta a los usuarios información sobre qué rutas puede realizar para conocer los sitios más emblemáticos de Madrid, información sobre donde almorzar o cenar mientras hacen esta ruta, u obtener una estimación del tiempo que les va a llevar realizarla. Por último, decir que permite obtener información sobre los distintos barrios de Madrid, así como los aspectos más relevantes de cada uno de ellos.

2.4.2 Mapa de Madrid offline

Mapa de Madrid offline [22] es una aplicación disponible para Android que permite mostrar ciertos elementos de la ciudad de Madrid de una forma gráfica. En cuanto a los elementos que esta aplicación es capaz de mostrar, decir que también es capaz de mostrar hospitales, bancos, restaurantes, así como tiendas o teatros.

La principal ventaja es la de no necesitar acceso a Internet para poder utilizarla pudiendo descargar toda la información relevante de Madrid en la aplicación. Otra ventaja que posee esta aplicación es la de poder buscar rutas no solo hasta los elementos anteriormente mencionados, sino que también permite buscar rutas hasta direcciones en concreto.

2.4.3 Google Maps

Google Maps [23] es una aplicación desarrollada por Google que busca ofrecer a los usuarios información sobre la ubicación en la que se encuentran. Gracias a esta aplicación, un usuario puede ubicarse independientemente del punto geográfico en el que se encuentre. Google Maps [48] viene ya descargada en todos los dispositivos Android, por lo que el volumen de usuarios a los que llega es bastante más elevado.

Permite conocer información sobre sitios de interés cercanos al usuario, así como poder conocer las opiniones y valoraciones que otros usuarios hacen sobre estos. Además, gracias a esta aplicación, se puede obtener la ruta hasta cualquier punto del mapa, así como una estimación del tiempo de llegada teniendo en cuenta la situación del tráfico actual.

2.4.4 Foursquare

Foursquare [24] es una aplicación basada en la geolocalización aplicada a las redes sociales. Se caracteriza porque utiliza mapas digitales para poder conocer qué centros, negocios o locales de interés hay cercanos respecto de una ubicación.

A la hora de utilizar este sistema, se pueden desempeñar dos roles distintos, el rol de usuario, y el rol de profesional o dueño de un establecimiento. En cuanto a las tareas que un usuario puede realizar se tienen las siguientes: encontrar, señalar y recomendar lugares específicos permitiendo interactuar con otros usuarios por medio de mensajes, valoraciones o sugerencias. En cuanto a las tareas de profesional se encuentran: dar de alta sus propios negocios y crear promociones para fomentar la participación

Un aspecto diferenciador de este proyecto frente a los demás, es el hecho de premiar a los usuarios. La aplicación se concibe como un juego en el que los usuarios más activos son premiados con descuentos en distintos servicios.

2.4.5 Yelp

Yelp [25] es una herramienta que surge con el fin de dar a conocer a las personas los negocios de su ciudad. Aprovecha el uso de la geolocalización para mostrar al usuario los negocios más cercanos.

Permite dar a conocer los pequeños negocios ya que los usuarios pueden realizar valoraciones de estos. Al igual que Foursquare, permite crear cuentas para empresas desde la que poder realizar ofertas y conseguir entablar relación con nuevos clientes.

Como aspecto diferenciador, hay que decir que este proyecto fomenta la interrelación de los usuarios, no solo a través de la aplicación, si no también organizando eventos semanales.

Capítulo 3. Desarrollo del proyecto

3.1 Marco de trabajo

SCRUM [45] es el marco de trabajo que se ha usado para realizar este proyecto, aunque, al ser solo uno el integrante del grupo se ha tenido que adaptar este marco de trabajo en algunos aspectos.

Para la realización de este proyecto han sido necesarias tres iteraciones, en las que, al principio de cada una, se producía una reunión con el tutor para determinar qué requisitos implementar. La iteración se acababa comentando los avances producidos. Las tres iteraciones han sido las siguientes:

1. Durante la primera iteración, se realizó una búsqueda de repositorios de datos abiertos de los que poder recuperar información relativa a los elementos de interés cultural y el desarrollo de una aplicación Android en la que los usuarios fueran capaces de gestionar su propia cuenta.
2. Durante la segunda iteración, se completó la demás funcionalidad de la aplicación Android con datos falseados. Los usuarios eran capaces de usar la aplicación, pero la información que se le mostraba era de demostración.
3. Durante la tercera iteración, la aplicación Android ya permitía trabajar con información real de los elementos de interés cultural de la ciudad de Madrid y se realizó una revisión global de la aplicación para la búsqueda de errores que se pudiesen existir para poder solucionarlos.

3.2 Especificación de requisitos

A continuación, se presentan los actores identificados que interactuarán con el sistema, así como los casos de uso que se han implementado.

3.2.1 Actores del sistema

Los actores que se han detectado han sido dos:

1. Actor sistema. Este actor representa al sistema. Es el actor encargado de buscar los elementos de interés cultural de la ciudad de Madrid, así como de determinar cuáles son los que se encuentran cerca de los usuarios.
2. Actor usuario. Este actor representa al usuario final de la aplicación.

3.2.2 Casos de uso

En la Figura 2 se muestra el diagrama de casos de uso del sistema. Los casos de uso se han separado en paquetes atendiendo al actor que los realiza.

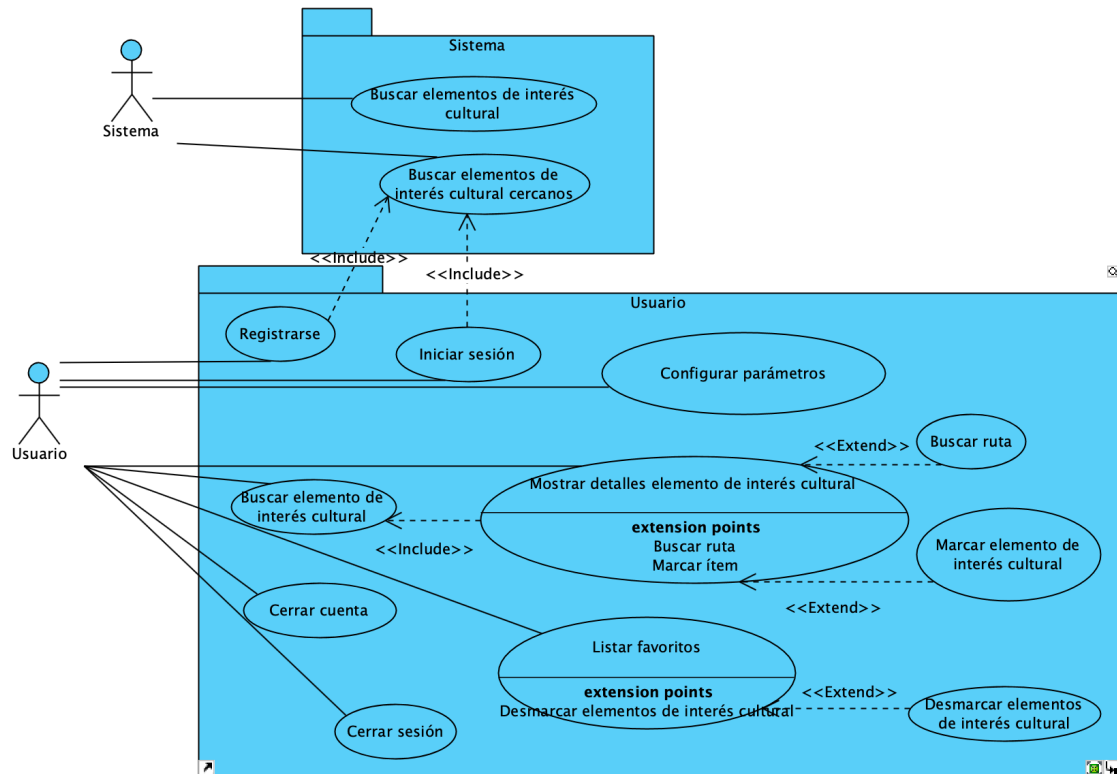


Figura 2 Casos de uso

A continuación, se muestran los casos de uso definidos agrupados de acuerdo con el actor que los realiza.

3.2.2.1 Casos de uso del sistema

UC-0001	Buscar elementos de interés cultural
Versión	1.0 (04/01/2019)
Autores	Juan Bautista Arellano Bruno
Fuentes	Cliente
Dependencias	Ninguno
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando se deba buscar información relativa a los elementos de interés cultural que se almacenan en el sistema
Precondición	Los datos de los elementos de interés cultural no están almacenados en el sistema
	Paso Acción

Secuencia normal	1	El sistema <i>descarga los datos de los elementos de interés cultural de fuente de datos abiertos</i>
	2	El sistema <i>almacena los datos de los elementos de interés cultural descargados en el formato deseado</i>
Postcondición	Los datos de los elementos de interés cultural quedan almacenados en el sistema	
Excepciones	Paso	Acción
	1	Si se produce un error, el sistema <i>notifica al administrador del sistema que se ha producido un error durante el proceso</i> , a continuación, este caso de uso queda sin efecto
	2	Si se produce un error, el sistema <i>notifica al administrador del sistema de que se ha producido un error durante el proceso</i> , a continuación, este caso de uso queda sin efecto
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	1 veces por semana(s)	
Importancia	vital	
Urgencia	inmediatamente	
Estado	pendiente de verificación	
Estabilidad	alta	
Comentarios	Ninguno	

UC-0002	Buscar elementos de interés cultural cercanos	
Versión	1.0 (04/01/2019)	
Autores	Juan Bautista Arellano Bruno	
Fuentes	Cliente	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>se deba de buscar elementos de interés cultural cercanos a la ubicación del usuario</i> o durante la realización de los siguientes casos de uso: [UC-0010] Registrarse , [UC-0011] Iniciar sesión	
Precondición	El usuario tiene activada la geolocalización del dispositivo activada	
Secuencia normal	Paso	Acción
	1	El sistema <i>recibe la ubicación del usuario y la distancia límite que el usuario marca para conocer los ítems cercanos</i>
	2	El sistema <i>busca todos los elementos de interés cultural cercanos al usuario</i>
	3	El sistema <i>retorna los elementos de interés cultural encontrados al usuario</i>
Postcondición	El usuario recibe los elementos de interés cultural cercanos a su ubicación	
Excepciones	Paso	Acción
	2	Si se produce un error, el sistema <i>notifica al usuario que no se han podido buscar elementos de interés cultural cercanos</i> , a continuación, este caso de uso queda sin efecto

Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	1 veces por minuto(s)	
Importancia	vital	
Urgencia	inmediatamente	
Estado	pendiente de verificación	
Estabilidad	alta	
Comentarios	Ninguno	

3.2.2.2 Casos de uso del usuario

UC-0003	Buscar elemento de interés cultural	
Versión	1.0 (04/01/2019)	
Autores	Juan Bautista Arellano Bruno	
Fuentes	Cliente	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desea buscar un elemento de interés cultural</i>	
Precondición	El usuario desea buscar información sobre un determinado elemento de interés cultural	
Secuencia normal	Paso	Acción
	1	El actor Usuario (ACT-0002) introduce el nombre del elemento de interés cultural que desea buscar
	2	El sistema busca la información referente a dicho elemento de interés cultural
	3	Se realiza el caso de uso Mostrar detalles elemento de interés cultural (UC-0004)
Postcondición	El sistema da a conocer la información solicitada al usuario	
Excepciones	Paso	Acción
	2	Si no existe dicho elemento de interés cultural o se produce un error durante este proceso, el sistema notifica al usuario de que no se ha podido buscar la información referente a dicho elemento de interés cultural, a continuación, este caso de uso continúa
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	3 veces por día(s)	
Importancia	vital	
Urgencia	inmediatamente	
Estado	pendiente de verificación	
Estabilidad	alta	
Comentarios	Ninguno	

UC-0004	Mostrar detalles elemento de interés cultural	
Versión	1.0 (04/01/2019)	
Autores	Juan Bautista Arellano Bruno	
Fuentes	Cliente	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desea conocer más información sobre un determinado elemento de interés cultural</i> o durante la realización de los siguientes casos de uso: [UC-0003] Buscar elemento de interés cultural	
Precondición	El usuario indica el elemento de interés cultural sobre el que desea conocer más información	
Secuencia normal	Paso	Acción
	1	El actor Usuario (ACT-0002) indica el elemento de interés cultural sobre el que desea conocer más detalles
	2	El sistema <i>comprueba que existe el elemento de interés cultural</i>
	3	El sistema <i>busca los detalles del elemento de interés cultural</i>
	4	El sistema <i>muestra los detalles recuperados del elemento de interés cultural</i>
	5	Si se desea marcar el elemento de interés cultural como favorito, se realiza el caso de uso Marcar elemento de interés cultural (UC-0005)
	6	Si el usuario indica que desea conocer la ruta al elemento de interés cultural, se realiza el caso de uso Buscar ruta (UC-0008)
Postcondición	El sistema muestra los detalles del elemento de interés cultural al usuario	
Excepciones	Paso	Acción
	2	Si <i>no existe el elemento de interés cultural indicado</i> , el sistema <i>notifica al usuario de que no se ha podido recuperar la información de dicho elemento de interés cultural</i> , a continuación, este caso de uso queda sin efecto
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	5 veces por semana(s)	
Importancia	vital	
Urgencia	inmediatamente	
Estado	pendiente de verificación	
Estabilidad	alta	
Comentarios	Ninguno	

UC-0006	Listar favoritos	
Versión	1.0 (04/01/2019)	
Autores	Juan Bautista Arellano Bruno	

Fuentes	Cliente	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desea visualizar los elementos de interés cultural que tiene marcados como favoritos</i> o durante la realización de los siguientes casos de uso: [UC-0007] Desmarcar elemento de interés cultural	
Precondición	El usuario desea listar los elementos de interés cultural que tiene marcados como favoritos	
Secuencia normal	Paso	Acción
	1	El actor Usuario (ACT-0002) indica que desea listar los elementos de interés cultural marcados como favoritos
	2	El sistema recupera los elementos de interés cultural que el usuario tiene marcados como favoritos
	3	El sistema muestra un listado con los elementos de interés cultural marcados por el usuario como favoritos
Postcondición	El sistema muestra un listado con los elementos de interés cultural marcados como favoritos	
Excepciones	Paso	Acción
	2	Si se produce un error, el sistema muestra el error producido durante el proceso al usuario, a continuación, este caso de uso queda sin efecto
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	1 veces por semana(s)	
Importancia	quedaría bien	
Urgencia	puede esperar	
Estado	pendiente de verificación	
Estabilidad	alta	
Comentarios	Ninguno	

UC-0005	Marcar elemento de interés cultural	
Versión	1.0 (04/01/2019)	
Autores	Juan Bautista Arellano Bruno	
Fuentes	Cliente	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desea marcar un elemento de interés cultural como favorito</i> o durante la realización de los siguientes casos de uso: [UC-0004] Mostrar detalles elemento de interés cultural	
Precondición	Se ha mostrado los detalles de un elemento de interés cultural	
	Paso	Acción

Secuencia normal	1	El actor Usuario (ACT-0002) indica que desea marcar el elemento de interés cultural como favorito
	2	El sistema <i>marca el elemento de interés cultural como favorito</i>
Postcondición	El elemento de interés cultural queda marcado como favorito para el usuario	
Excepciones	Paso	Acción
	-	-
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	2 veces por semana(s)	
Importancia	quedaría bien	
Urgencia	puede esperar	
Estado	pendiente de verificación	
Estabilidad	alta	
Comentarios	Ninguno	

UC-0007	Desmarcar elemento de interés cultural	
Versión	1.0 (04/01/2019)	
Autores	Juan Bautista Arellano Bruno	
Fuentes	Cliente	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>desmarcar un elemento de interés cultural como favorito</i>	
Precondición	El elemento de interés cultural ha sido marcado previamente como favorito	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Listar favoritos (UC-0006)
	2	El actor Usuario (ACT-0002) indica que elemento de interés cultural desea <i>desmarcar como favorito</i>
	3	El sistema <i>elimina del listado de favoritos el elemento de interés cultural indicado por el usuario</i>
Postcondición	El elemento de interés cultural deja de estar marcado como favorito	
Excepciones	Paso	Acción
	3	Si no se pueda eliminar el elemento de interés cultural del listado de favoritos, el sistema <i>notifica al usuario de que se ha producido un error durante el proceso, a continuación, este caso de uso queda sin efecto</i>
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	1 veces por semana(s)	
Importancia	quedaría bien	
Urgencia	puede esperar	

Estado	pendiente de verificación
Estabilidad	alta
Comentarios	Ninguno

UC-0008	Buscar ruta	
Versión	1.0 (04/01/2019)	
Autores	Juan Bautista Arellano Bruno	
Fuentes	Cliente	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desea conocer la ruta hasta el elemento de interés cultural seleccionado desde el punto en el que se encuentra</i> o durante la realización de los siguientes casos de uso: [UC-0004] Mostrar detalles elemento de interés cultural	
Precondición	El usuario tiene activada la geolocalización en su dispositivo	
Secuencia normal	Paso	Acción
	1	El actor Usuario (ACT-0002) indica que desea saber como llegar hasta un determinado elemento de interés cultural desde el punto en el que se encuentra
	2	El sistema calcula la ruta hasta el elemento de interés cultural desde el punto en el que se encuentra
	3	El sistema muestra la ruta hasta el elemento de interés cultural desde el punto en el que se encuentra el usuario
Postcondición	Se muestra la ruta hasta el museo/monumento desde el punto en el que se encuentra el usuario	
Excepciones	Paso	Acción
	2	Si no se puede calcular la ruta hasta el elemento de interés cultural seleccionado, el sistema notifica al usuario de que se ha producido un error durante el proceso, a continuación, este caso de uso queda sin efecto
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	1 veces por día(s)	
Importancia	quedaría bien	
Urgencia	puede esperar	
Estado	pendiente de verificación	
Estabilidad	alta	
Comentarios	Ninguno	

UC-0009	Configurar parámetros	
Versión	1.0 (04/01/2019)	
Autores	Juan Bautista Arellano Bruno	

Fuentes	Cliente	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desea realizar algún cambio de los parámetros que afectan al comportamiento de la aplicación</i>	
Precondición	El usuario indica que desea cambiar los parámetros del sistema	
Secuencia normal	Paso	Acción
	1	El actor Usuario (ACT-0002) indica que desea realizar cambios en los parámetros de la aplicación
	2	El sistema <i>muestra los valores de los parámetros de la aplicación</i>
	3	El actor Usuario (ACT-0002) modifica el valor de los parámetros del sistema
	4	El actor Usuario (ACT-0002) desea guardar los nuevos valores de los parámetros del sistema
	5	El sistema <i>comprueba la validez de los nuevos valores de los parámetros del sistema</i>
	6	Si los datos son válidos, el sistema guarda los nuevos valores de los parámetros
Postcondición	Se realizan los cambios realizados por el usuario	
Excepciones	Paso	Acción
	2	Si no se puedan mostrar los parámetros, el sistema notifica al usuario de que se ha producido un error durante el proceso, a continuación, este caso de uso queda sin efecto
	5	Si los nuevos valores introducidos no son válidos, el sistema notifica al usuario de un error durante el proceso, a continuación, este caso de uso queda sin efecto
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	1 veces por día(s)	
Importancia	vital	
Urgencia	inmediatamente	
Estado	pendiente de verificación	
Estabilidad	alta	
Comentarios	Ninguno	

UC-0010	Registrarse
Versión	1.0 (12/01/2019)
Autores	Juan Bautista Arellano Bruno
Fuentes	Cliente
Dependencias	Ninguno
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desea registrarse en el sistema</i>
Precondición	El usuario no está registrado en el sistema

Secuencia normal	Paso	Acción
	1	El actor Usuario (ACT-0002) indica que desea registrarse en el sistema
	2	El actor Usuario (ACT-0002) introduce los datos necesarios para poder registrarse
	3	El sistema valida los datos introducidos por el usuario
	4	Si los datos son válidos, el sistema registra al usuario en el sistema
	5	Se realiza el caso de uso Buscar elementos de interés cultural cercanos (UC-0002)
Postcondición	El usuario se registra en el sistema	
Excepciones	Paso	Acción
	3	Si Los datos introducidos por el usuario no son válidos o ya existe un usuario con los datos introducidos, el sistema notifica al usuario de que los datos no son válidos, a continuación, este caso de uso queda sin efecto
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	1 veces por año(s)	
Importancia	vital	
Urgencia	inmediatamente	
Estado	pendiente de verificación	
Estabilidad	alta	
Comentarios	Ninguno	

UC-0011	Iniciar sesión	
Versión	1.0 (12/01/2019)	
Autores	Juan Bautista Arellano Bruno	
Fuentes	Cliente	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>El usuario desea iniciar sesión en el sistema</i>	
Precondición	El usuario desea iniciar sesión en el sistema	
Secuencia normal	Paso	Acción
	1	El actor Usuario (ACT-0002) indica que desea iniciar sesión en el sistema
	2	El actor Usuario (ACT-0002) introduce los datos necesarios para iniciar sesión en el sistema
	3	El sistema valida los datos introducidos por el usuario
	4	Si los datos introducidos por el usuario son válidos, el actor Usuario (ACT-0002) inicia sesión en el sistema
	5	Se realiza el caso de uso Buscar elementos de interés cultural cercanos (UC-0002)
Postcondición	El usuario inicia sesión sesión en sistema	
Excepciones	Paso	Acción

	3	Si los datos introducidos por el usuario no son válidos, el sistema notifica al usuario de que los datos introducidos no son válidos, a continuación, este caso de uso queda sin efecto
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	1 veces por semana(s)	
Importancia	vital	
Urgencia	Inmediatamente	
Estado	pendiente de verificación	
Estabilidad	Alta	
Comentarios	Ninguno	

UC-0012	Cerrar sesión	
Versión	1.0 (12/01/2019)	
Autores	Juan Bautista Arellano Bruno	
Fuentes	Cliente	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>el usuario desea cerrar una sesión abierta</i>	
Precondición	El usuario posee una sesión abierta	
Secuencia normal	Paso	Acción
	1	El actor Usuario (ACT-0002) indica que desea cerrar sesión
	2	El sistema <i>cierra la sesión del usuario</i>
Postcondición	El usuario cierra la sesión	
Excepciones	Paso	Acción
	-	-
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	1 veces por semana(s)	
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	pendiente de verificación	
Estabilidad	Alta	
Comentarios	Ninguno	

UC-0013	Cerrar cuenta	
Versión	1.0 (12/01/2019)	

Autores	Juan Bautista Arellano Bruno	
Fuentes	Cliente	
Dependencias	Ninguno	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando <i>El usuario desea cancelar su cuenta</i>	
Precondición	El usuario está registrado en el sistema	
Secuencia normal	Paso	Acción
	1	El actor Usuario (ACT-0002) indica que desea cerrar la cuenta que posee
	2	El sistema cierra la cuenta que posee el usuario
Postcondición	El sistema cierra la cuenta del usuario	
Excepciones	Paso	Acción
	-	-
Rendimiento	Paso	Tiempo máximo
	-	-
Frecuencia esperada	1 veces por año(s)	
Importancia	Vital	
Urgencia	Inmediatamente	
Estado	pendiente de verificación	
Estabilidad	Alta	
Comentarios	Ninguno	

3.3 Arquitectura

La aplicación desarrollada sigue la arquitectura cliente-servidor [46], de la cual, se puede apreciar un esquema de su funcionamiento en la Figura 3. En este proyecto existe un cliente (la aplicación Android) que demanda información al servidor sobre los elementos de interés cultural de Madrid.

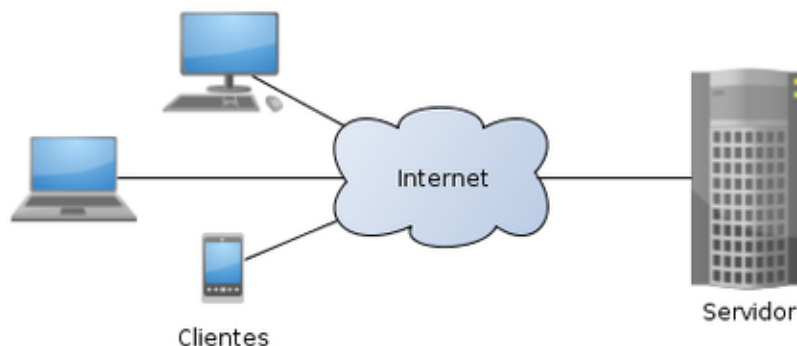


Figura 3 Arquitectura Cliente-Servidor

Se ha decidido utilizar este tipo de arquitectura porque presenta las siguientes ventajas:

1. Centralización del control. Los accesos, recursos e integridad de los datos con los que trabaja el sistema son controlados por el servidor de forma que un programa no autorizado pueda dañar el sistema.
2. Escalabilidad. Se puede aumentar la capacidad de los clientes y servidores por separado.
3. Fácil mantenimiento. Al estar distribuidas las funciones y responsabilidades entre varios ordenadores independientes, es posible reparar, trasladar o reemplazar un servidor sin la necesidad de que se vea afectado un cliente.

3.4 Modelo de datos

3.4.1 Modelo Entidad-Relación

En la Figura 4, se muestra el diagrama entidad relación del sistema en el que se muestra la información que se requiere almacenar respecto a los usuarios, los elementos de interés cultural de la ciudad de Madrid, así como la información sobre los elementos de interés cultural que cada usuario desea guardar.

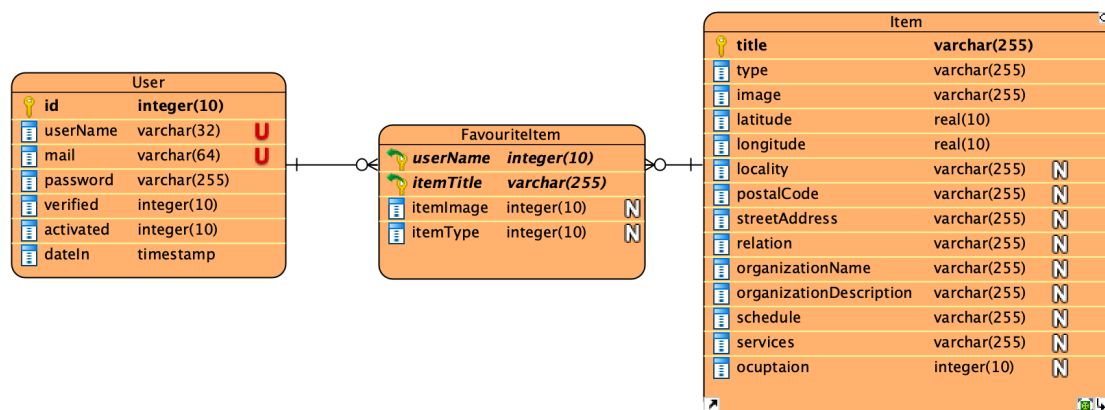


Figura 4 Modelo Entidad Relación

3.4.2 Base de datos

Para realizar la implementación del modelo entidad relación anterior se ha requerido la utilización tanto de modelos relacionales como de modelos no relacionales.

3.4.2.1 Base de datos relacional

Para almacenar información de los usuarios del sistema se ha usado una base de datos MySQL. Esta base de datos posee una única tabla en la que se almacena toda la información de los distintos usuarios. La estructura de la tabla es la siguiente:

1. Nombre de la tabla: "Users".
2. Columnas de la tabla:
 - a. id: Valor entero que se utiliza como identificador de cada uno de los usuarios.
 - b. userName: Cadena de texto que se utiliza para referirse al usuario.
 - c. mail: Cadena de texto que almacena la dirección de correo con la que se registra el usuario del sistema.
 - d. password: Cadena de texto que almacena el hash de la contraseña del usuario.
 - e. verified: Valor entero, el cual solo puede ser 0 o 1 que se utiliza para saber si la cuenta del usuario está o no verificada.
 - f. activated: Valor entero, el cual solo puede ser 0 o 1 que se utiliza para saber si la cuenta del usuario está o no activa.
 - g. dateIn: Valor de tipo timestamp que almacena la fecha en la que el usuario se registró en el sistema.

3.4.2.2 Base de datos NoSQL

Se ha usado una base de datos documental de tipo MongoDB para almacenar toda la información de los elementos de interés cultural de los usuarios, así como de aquellos elementos que guarda cada usuario como favorito. A continuación, se detallan las colecciones que se han utilizado para almacenar dicha información.

1. Colección "Items". Contiene toda la información relativa a los elementos de interés cultural de Madrid. Los campos que posee esta colección son los siguientes:
 - a. title: Cadena de texto que contiene el nombre del elemento de interés cultural.
 - b. type: Cadena de texto que contiene el tipo de elemento de cultural del que se trata.
 - c. image: Cadena de texto que contiene un enlace a una imagen del elemento de interés cultural.
 - d. relation: Cadena de texto que contiene el enlace en el que se puede obtener más información del elemento de interés cultural.
 - e. location: Campo que contiene la información relativa a la localización del elemento de interés cultural. En concreto posee los siguientes campos:
 - i. latitude: Número real que contiene la latitud en la que se encuentra el elemento de interés cultural.
 - ii. longitude: Número real que contiene la longitud en la que se encuentra el elemento de interés cultural.
 - f. address: Campo que contiene la información relativa a la dirección en la que se encuentra el elemento de interés cultural. En concreto posee los siguientes campos:

- i. locality: Cadena de texto que contiene la localidad en la que se encuentra el elemento de interés cultural.
 - ii. postal-code: Cadena de texto que contiene el código postal en el que se encuentra el elemento de interés cultural.
 - iii. street-address: Cadena de texto que contiene la dirección, en particular, calle y número, de la ubicación del elemento de interés cultural.
 - g. organization: Campo contiene información de la organización del elemento de interés cultural. En concreto, posee los siguientes campos:
 - i. organization-name: Cadena de texto que contiene el nombre de la organización.
 - ii. organization-description: Cadena de texto que contiene una descripción del elemento de interés cultural.
 - iii. schedule: Cadena de texto que contiene el horario de apertura del elemento de interés cultural.
 - iv. services: Cadena de texto que contiene los servicios que ofrece el elemento de interés cultural.
 - h. occupation: Vector de valores que contiene información sobre la afluencia de visitas que recibe el elemento de interés cultural cada día de la semana. Decir que los datos están falseados.
2. Colección “Favourites” Contiene la información de los elementos de interés cultural que los usuarios marcan como favoritos. Los campos que posee esta colección son los siguientes:
- a. userName. Cadena de texto que contiene el nombre de usuario que ha añadido un elemento de interés cultural como favorito.
 - b. favourites. Vector de elementos de interés cultural que un usuario ha marcado como favorito. Cada uno de los elementos de este vector posee los siguientes campos:
 - i. name: Cadena de texto que contiene el nombre del elemento de interés cultural.
 - ii. type: Cadena de texto que contiene el tipo del elemento de interés cultural.
 - iii. image: Cadena de texto que contiene la dirección de la imagen asociada al elemento de interés cultural.

Capítulo 4. Diseño de la aplicación

En este capítulo se explicará las principales clases o scripts desarrolladas para el sistema. En la Figura 5 se muestra un diagrama de despliegue en el que se puede apreciar como la aplicación Android, a través de Internet puede interactuar con los ficheros PHP y Python almacenados en el servidor, y los cuales interactúan con las bases de datos correspondientes. A continuación, se va a mostrar algunos ejemplos de cómo interactúan estos ficheros.

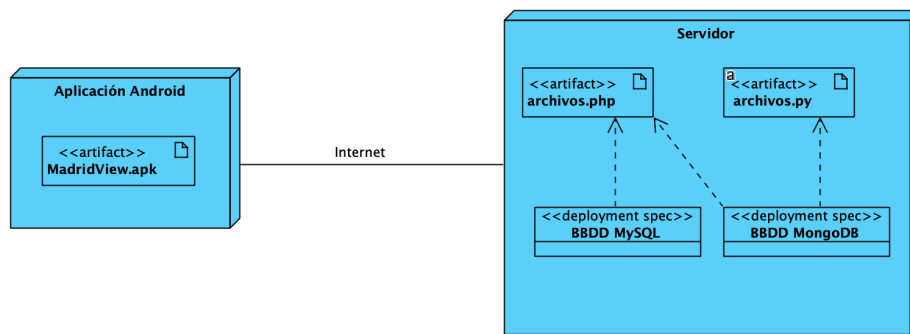


Figura 5 Modelo de despliegue

4.1 Registro de usuarios

El fichero de la aplicación encargado de realizar el registro de usuarios es el llamado "RegisterActivity.java".

```

if(!error){
    HashMap<String, String> params = new HashMap<>();
    params.put("userName", userNameInput.getText().toString());
    params.put("mail", mailInput.getText().toString());
    params.put("password1", passwordInput1.getText().toString());
    params.put("password2", passwordInput2.getText().toString());

    HttpHandler.post(getApplicationContext(), url: ServerDataUtils.SERVER_URL + "/pages/php/checkSignUp.php", params, new VolleyCallback() {
        @Override
        public void onSuccessResponse(JSONObject response) {
            try {
                if (response != null) {
                    if (response.getInt( name: "success") == 1) {

                        User user = User.getInstance();
                        user.setUserName(getApplicationContext(), response.getString( name: "userName"));
                        user.setMail(getApplicationContext(), response.getString( name: "mail"));
                        user.setUserDataIn(getApplicationContext(), response.getString( name: "dateIn"));
                        user.setActivated(getApplicationContext(), response.getInt( name: "activated"));
                        user.setVerified(getApplicationContext(), response.getInt( name: "verified"));

                        Toast.makeText(getApplicationContext(), "Register completed", Toast.LENGTH_LONG).show();
                        Intent intent = new Intent(getApplicationContext(), HomeActivity.class);
                        startActivity(intent);
                    }
                    else{
                        switch (response.getInt( name: "code")){
                            case ServerDataUtils.INTERNAL_ERROR:
                                Toast.makeText(getApplicationContext(), "Unknown error on server", Toast.LENGTH_LONG).show();
                                break;
                            case ServerDataUtils.USERNAME_ALREADY_EXISTS:
                                userNameInput.setError("User name not available");
                                break;
                            case ServerDataUtils.MAIL_ALREADY_EXISTS:
                                mailInput.setError("Mail not available");
                                break;
                            case ServerDataUtils.DIFFERENT_PASSWORDS_ERROR:
                                passwordInput2.setError("Passwords do not match");
                                break;
                            default:
                                Toast.makeText(getApplicationContext(), "Unknown error on server", Toast.LENGTH_LONG).show();
                        }
                    }
                }
            } catch (JSONException e){
                Toast.makeText(getApplicationContext(), "Unknown error on server", Toast.LENGTH_LONG).show();
                Log.d(TAG, e.getMessage());
            }
        }
    });
}

```

Figura 6 Registro de Usuarios

En la Figura 6 se puede apreciar el código que implementa el alta de usuarios. Si no hay ningún error a la hora de leer la información requerida para realizar el registro del usuario, se procede a realizar una petición HTTP al servidor para dar de alta al usuario en el sistema haciendo uso de la librería Volley. El fichero en particular que se encarga de realizar este proceso es el fichero “checkSignUp.php”. En la Figura 7 se muestra el código que se encarga de almacenar la información del usuario en la base de datos relacional. Tras comprobar que la información recibida del usuario es correcta y que no existe ningún usuario ya con dicha información, se procede al registro del usuario en la base de datos. Posteriormente, se procede a tomar el identificador del nuevo usuario para generar un enlace específico para que pueda verificar su cuenta gracias al envío de un correo de bienvenida que se manda de forma automática. Para finalizar, se envía la información necesaria al usuario para que pueda iniciar sesión en la aplicación, y por tanto hacer uso de ella.

```

$hash = password_hash($password1, PASSWORD_DEFAULT);

$stmt = "INSERT INTO Users (userName, mail, password) VALUES (?, ?, ?)";
$sentence = $link->stmt_init();

if(!$sentence->prepare($stmt)){
    mysqli_close($link);
    returnError($fileName, "Sorry, can not prepare sentence to register the new user", 1);
}
else{
    $sentence->bind_param("sss", $userName, $mail, $hash);
    $sentence->execute();
    $sentence->close();

    $stmt = "SELECT * FROM Users WHERE userName=?";
    $sentence = $link->stmt_init();

    if(!$sentence->prepare($stmt)){
        mysqli_close($link);
        returnError($fileName, "Can not get user info from database", 1);
    }else{
        $sentence->bind_param('s', $userName);
        $sentence->execute();
        $result = $sentence->get_result();
        $sentence->close();

        mysqli_close($link);

        if($result->num_rows != 1){
            returnError($fileName, "Can not find data from user already registered", 2);
        }
        else{
            $userID = -1;

            while($row = $result->fetch_array(MYSQLI_ASSOC)){
                $userID = $row['id'];
                break;
            }

            $userLink = "https://".$ip."/pages/php/verifyAccount.php?usr=".openssl_encrypt($userID, CIPHER, CIPHER_KEY);

            $body = "<h1>Bienvenido ". $userName."</h1>
            <p>Madrid View le da las gracias por utilizar esta aplicación para conocer la ciudad de Madrid.</p>
            <p>Madrid View le comunica que está a su disposición ante cualquier duda que pueda tener.</p>
            <p>Para verificar su cuenta, pulse en el enlace que aparece a continuación.</p>
            <a href='". $userLink. "'>Enlace</a>
            <p>Un saludo, Juan Arellano.</p>";

            if(!sendMail($mail, $userName, "Bienvenido a Madrid View", $body)){
                returnError($fileName, "Can not send verification mail", 1);
            }
            else{
                $datas = array("userName" => $row["userName"], "mail"=>$row["mail"], "verified"=>$row["verified"], "activated"=>$row["activated"], "
                dateIn"=>$row["dateIn"]);
                $array = array("success" => 1, "datas" => $datas);
                echo json_encode($array);
            }
        }
    }
}

```

Figura 7 Registro de usuario en la base de datos

4.2 Búsqueda de elementos de interés cultural cercanos

En este apartado se muestra cómo se relacionan diferentes ficheros para poder mostrar a los usuarios aquellos elementos de interés cultural que se encuentran cerca. El fichero de la aplicación encargado de tomar la localización del dispositivo es el fichero “LocationService.java”. En la Figura 8 se puede apreciar el código que se encarga de realizar la petición HTTP a una Google Cloud Function. En dicha petición se puede ver que se proporciona la longitud, latitud y distancia a la que encontrar los elementos de interés cultural, así como qué tipos de elementos buscar y cuáles no.


```

AppSettings appSettings = AppSettings.getInstance();

JSONObject params = new JSONObject();

params.put( name: "latitude", value: ""+Double.toString(location.getLatitude()));
params.put( name: "longitude", value: ""+ Double.toString(location.getLongitude()));
params.put( name: "distance", Float.toString(appSettings.getCloseDistance(getApplicationContext())));
//params.put("distance", Float.toString(500));

params.put( name: "searchChurch", Integer.toString(appSettings.searchChurchesEnabled(getApplicationContext())? 1:0));
params.put( name: "searchMuseum", Integer.toString(appSettings.searchMuseumsEnabled(getApplicationContext())? 1:0));
params.put( name: "searchMonument", Integer.toString(appSettings.searchMonumentsEnabled(getApplicationContext())? 1:0));
params.put( name: "searchTouristOffice", Integer.toString(appSettings.searchTouristOfficeEnabled(getApplicationContext())? 1:0));
params.put( name: "searchPark", Integer.toString(appSettings.searchParksEnabled(getApplicationContext())? 1:0));

HttpHandler.post(getApplicationContext(), ServerDataUtils.SEARCH_CLOSE_ITEMS_URL, params, new VolleyCallback() {
    @Override
    public void onSuccessResponse(JSONObject response) {
        try{
            if(response!=null){
                int code = response.getInt( name: "success");

                if(code!=1){
                    Toast.makeText(getApplicationContext(), response.getString( name: "error"),Toast.LENGTH_LONG).show();
                }
                else{
                    Publisher.publish(getApplicationContext(), AppUtils.CLOSE_ITEMS_FOUND_ACTION, response);
                }
            }
        }catch(JSONException e){
            e.printStackTrace();
        }
    }

    @Override
    public void onFailResponse(String error) { Log.e(TAG, error); }
});

```

Figura 8 Petición de elementos de interés cultural cercanos

Dicha petición, se recoge en una Google Cloud Function en la que tras comprobar que la información recibida es correcta, recoge aquellos elementos de interés cultural que se encuentran cerca del usuario y que son del tipo que el usuario desea encontrar.

```
def searchCloseItems(longitude, latitude, distance, searchMonument, searchMuseum, searchPark, searchTouristOffice, searchChurch):

    host = "35.230.149.72"
    user = "mongoUser"
    password = "MadridView1234"
    madridViewDBName = "madridViewDB"
    uri = "mongodb://"+user+":"+password+"@"+host+"/"+madridViewDBName

    client = pymongo.MongoClient(uri)
    itemsCollectionName = "Items"

    try:

        info = client.server_info()
        db = client[madridViewDBName]
        itemsCollection = db[itemsCollectionName]

        items = []
        #db.Items.find({"type":{"$in":["Monumento", "Parque"]}}).count();
        types = []

        if searchMonument==1:
            types.append("Monumento")

        if searchMuseum==1:
            types.append("Museo")

        if searchChurch==1:
            types.append("Iglesia católica")

        if searchPark==1:
            types.append("Parque")

        if searchTouristOffice==1:
            types.append("Oficina de Turismo")

        for item in itemsCollection.find({"type":{"$in":types}}, {"_id":0, "relation":0, "address":0, "organization":0}):

            if 'location' in item:
                if (mpu.haversine_distance((longitude, latitude), (item['location']['longitude'], item['location']['latitude'])*1000) <= distance:
                    items.append(item)

        client.close()

        return items, 0

    except pymongo.errors.ServerSelectionTimeoutError as e:
        print("Can not connect to mongo database")
        return None, -1
```

Figura 9 Obtener los elementos de interés cultural cercanos

En la Figura 9 puede apreciar el código de la Google Cloud Function que se encarga de recuperar los elementos de interés cultural cercanos. Para ello se conecta a la base de datos MongoDB y haciendo uso de la librería MPU de Python, se recuperan los elementos de interés cultural cercanos para posteriormente enviárselos como respuesta al usuario que realizó la petición. Una vez la aplicación tiene acceso a la información de respuesta, es capaz de mostrarlo gráficamente al usuario en el mapa de Madrid. En la Figura 10 se puede apreciar cómo se le mostrarían al usuario los elementos de interés cultural cercanos.

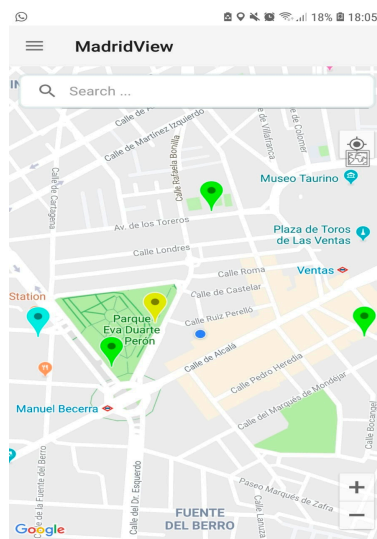


Figura 10 Aplicación con los elementos de interés cultural cercanos

4.3 Búsqueda de elementos de interés cultural de Madrid

En la Figura 11 se muestra el fichero “itemBrowser.py” que lleva a cabo la búsqueda de elementos de interés cultural de Madrid haciendo uso del repositorio de datos abiertos del ayuntamiento de Madrid. Dicho fichero, realiza una serie de peticiones al repositorio de datos abiertos del ayuntamiento de Madrid. Cada vez que se obtiene respuesta, se realiza una traducción para almacenar la información en el formato apropiado, de igual modo, se comprueba si dicho elemento ya existe en el sistema. En caso de que no sea así, se procede a buscar una imagen del elemento utilizando la herramienta de Google conocida como Google Custom Search. Finalmente se realiza la inserción de todos los elementos de un mismo tipo de forma conjunta.

```
for url, itemKind in zip(urls, itemKinds):
    print("Searching "+itemKind)
    try:
        items = []
        response = urllib.request.urlopen(url)
        if response.getcode() == 200:
            html = response.read()
            data = json.loads(html)
            for element in data["@graph"]:
                item = parseElementWithTemplate(element, template)
                if itemsCollection.count_documents({"title":item["title"]}) == 0:
                    item["type"] = itemKind
                    item["image"] = ""
                    result = googleCustomSearchImage(item["title"])
                    if result is not None:
                        item["image"] = result
                    items.append(item)
            if len(items) > 0:
                itemsCollection.insert_many(items)
        else:
            print("Something went wrong getting attributes")
    except Exception as e:
        raise e
```

Figura 11 Buscador de elementos de interés cultural de Madrid

Capítulo 5. Herramientas tecnológicas

A continuación, se describen las principales herramientas tecnológicas que se han utilizado en el desarrollo del proyecto.

5.1 Lenguajes de programación

5.1.1 Python

Python es un lenguaje de programación multiplataforma administrado por la Python Software Foundation, el cual, posee código de licencia de código abierto [26]. Este lenguaje es ideal para trabajar con grandes volúmenes de datos porque favorece su extracción y procesamiento, no solo con esto, para el ámbito científico, proporciona una gran cantidad de recursos matemáticos. Python, además del uso que se le quiera dar, cuenta con una amplia comunidad de usuarios que comparten constantemente sus conocimientos [27].

Algunas de las características más importantes de Python son [28]:

1. Lenguaje de propósito general con el que se busca desarrollar cualquier tipo de programa.
2. Lenguaje interpretado. El código se compila de manera transparente al programador.
3. Es orientado a objetos.
4. Posee una gran cantidad de funciones y de librerías gracias a las que poder tratar aspectos de programación específicos.

5.1.2 Java

Java es un lenguaje de programación orientado a objetos y una plataforma informática que nació con el objetivo de poder ser ejecutado en diversos sistemas operativos [29]. Java deriva de C, por lo que las reglas de sintaxis son muy similares a las de este lenguaje. Cuando se programa en Java, el código se escribe en archivos .java y luego se compilan, para así, generar el código byte en los ficheros .class, los cuales, contienen las instrucciones destinadas a ejecutarse en la máquina virtual de java (JVM), la cual, es un software específico de cada plataforma, surgiendo así el principio “Escrito una vez, ejecutado en cualquier lugar” del lenguaje Java [30]. Decir que Java es utilizado para crear aplicaciones y procesos en una gran cantidad de dispositivos distintos, se puede aplicar para el desarrollo de aplicaciones tanto en Smartphone como en electrodomésticos [29].

5.1.3 PHP

PHP (PHP: Hypertext preprocessor) es un lenguaje de código abierto adecuado para el desarrollo web [31].

Lo que distingue a PHP de otros lenguajes como puede ser por ejemplo Javascript, es que el código es ejecutado en el servidor, generando HTML y enviándoselo al cliente. El cliente así recibe el resultado de generar el script, sin la necesidad de conocer el código ejecutado.

PHP puede ser utilizado en diversos sistemas operativos y en la mayor parte de servidores de la actualidad. Permite utilizar la programación por procedimientos, programación orientada a objetos, o una mezcla de ambas. Una de las características más destacables de PHP es el soporte y documentación que ofrece para trabajar con múltiples bases de datos diferentes.

En cuanto a las aplicaciones más comunes que tiene este lenguaje, se tiene el desarrollo de scripts del lado del servidor, scripts desde línea de comandos, o bien, escribir aplicaciones de escritorio, aunque en menor parte que las otras dos aplicaciones anteriores.

5.2 Librerías utilizadas

5.2.1 PHPMailer

PHPMailer [32] es una clase destacada del ecosistema PHP que facilita el envío de email desde PHP, sobre todo, en correos especialmente complejos. Está disponible con licencia de libre distribución y uso. Esta clase se puede obtener a partir del siguiente enlace: <https://github.com/PHPMailer/PHPMailer>

5.2.2 Volley

Volley [33] es una librería desarrollada por Google para optimizar el envío de peticiones HTTP desde las aplicaciones Android a servidores externos. Permite liberar al desarrollador de la administración de hilos y procesos publicando fácilmente resultados en el hilo principal, facilitando así, la comunicación de las aplicaciones con Internet.

Posee varios componentes que permiten gestionar todas las peticiones que se realizan de una forma óptima. La gestión comienza con una cola de peticiones, la cual, recibe todas las peticiones que se van a realizar y las prioriza. A continuación, las peticiones son seleccionadas por un elemento llamado Cache Dispatcher. Este elemento permite comprobar si la respuesta de la petición actual puede ser obtenida de resultados previos de peticiones que ya se hayan realizado y se hayan guardado en caché. Si el resultado se puede recuperar, entonces se presenta la respuesta de la petición al hilo principal. En

caso de que no se pueda recuperar la respuesta, se envía la petición a la cola de conexiones pendientes. Por último, el Network Dispatcher, es el encargado de seleccionar las peticiones pendientes de la cola para realizar las respectivas transacciones Http con el servidor correspondiente. Si es necesario, la respuesta de las peticiones se almacena en caché y finalmente se publican en el hilo principal.

En la siguiente ilustración se puede apreciar un esquema sobre cómo funciona esta librería a más bajo nivel.

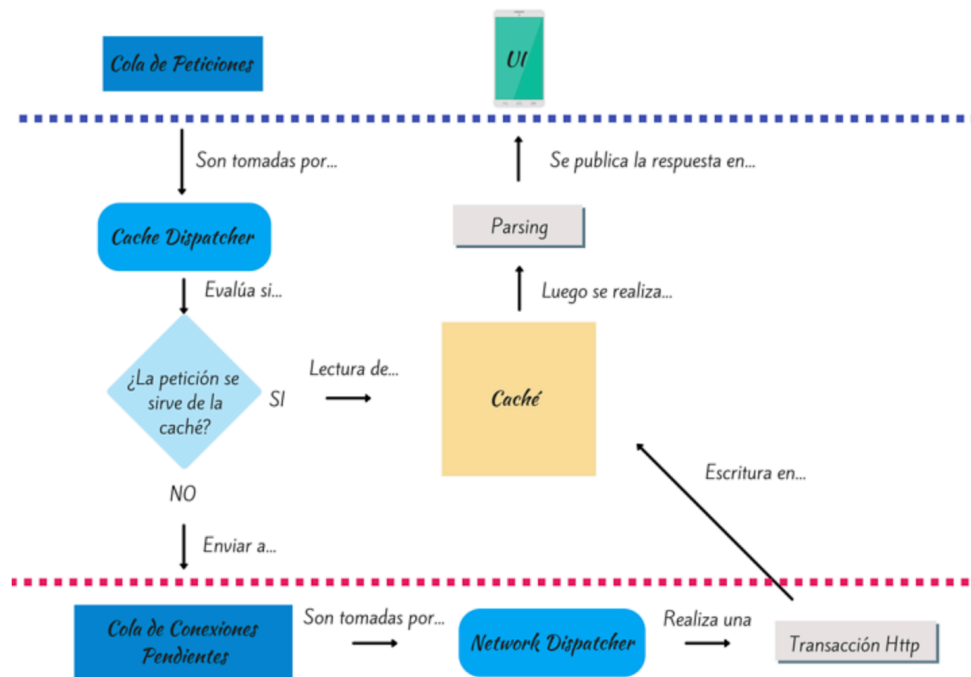


Figura 12 Funcionamiento de Volley

5.3 Entornos de desarrollo

5.3.1 Android Studio

Android es un sistema operativo móvil, basado en Linux, desarrollado para dispositivos móviles con pantalla táctil, así como pueden ser los teléfonos inteligentes, wearable o tablets.

Android fue desarrollado por Android Inc, compañía fundada en 2003 por Andy Rubin, Rich Miner, Nick Sears y Chris White en california. En un primer momento, esta compañía fue apoyada por Google, aunque, finalmente, fue comprada por Google. A pesar de todo, no fue hasta el año 2007 cuando este sistema operativo se presentaría al mercado [34].

La estructura del sistema operativo se basaba en un conjunto de aplicaciones que se ejecutan sobre una máquina virtual Dalvik con compilación en tiempo de ejecución,

aunque posteriormente se modificó, y esta fue sustituida por el entorno Android Runtime (ART).

Android Studio es el entorno de desarrollo (IDE) oficial para el desarrollo de aplicaciones Android basado en IntelliJ IDEA. Esta herramienta está disponible para múltiples plataformas, como puede ser Mac OS, Windows y GNU/Linux [35].

Además del editor de códigos que proporciona, Android Studio ofrece múltiples funciones que permiten aumentar la productividad a la hora de desarrollar aplicaciones Android:

1. Un sistema de compilación basado en gradle flexible.
2. Un emulador con varias funciones.
3. Instant run, que es la propiedad de poder aplicar cambios en la aplicación sin la necesidad de generar una nueva APK.
4. Gran cantidad de herramientas y frameworks de prueba.
5. Soporte incorporado para Google cloud Platform.

5.3.2 Google Cloud Platform

La computación en la nube, conocida también como Cloud Computing es un tipo de paradigma¹ que permite ofrecer servicios u ofertas de computación a través una red, por lo general, Internet [36].

Google Cloud es una plataforma de computación en la nube que contiene diversos servicios y herramientas para poder desarrollar aplicaciones garantizando una gran escalabilidad y seguridad [38].

El número de herramientas que presenta es elevado. En particular se tiene la plataforma “Compute Engine”, la cual permite dotar de poder computacional a máquinas en la nube. Esta plataforma se centra en tener una infraestructura como servicio, en la cual se tiene que configurar cada aspecto y hacer la gestión de recursos. Este servicio cobra por uso de recursos.

¹ Dentro de este tipo de paradigma, no todas las ofertas que se ofrecen son iguales. Se pueden distinguir tres tipos de ofertas diferentes [37]:

1. IaaS. Es un tipo de oferta en la que un proveedor proporciona a los clientes acceso a recursos de cálculo, por ejemplo, servidores o almacenamiento.
2. PaaS. Es una oferta en la que un proveedor proporciona a los clientes un entorno en la nube en el que pueden desarrollar, gestionar y distribuir sus propias aplicaciones sin la necesidad de preocuparse por la infraestructura subyacente ya que es el proveedor el que se encarga de esta.
3. SaaS. Es una oferta que proporciona acceso a los clientes a un software basado en la nube de un determinado proveedor. Los usuarios no tienen la necesidad de instalar ninguna aplicación en sus dispositivos, sino que las aplicaciones se encuentran ya en la nube. Los usuarios suelen hacer uso de las aplicaciones a través de un modelo de suscripción, de manera que no deben de gestionar el software si no que son los proveedores los que realizan esta tarea.

Otro servicio que se ha utilizado y que ofrece esta plataforma son las Cloud Functions [39], las cuales permiten ejecutar código en respuesta a ciertos eventos. El código se almacena y se ejecuta en un entorno administrado. Cloud Functions es un tipo de oferta PaaS que ofrece Google Cloud.

5.4 Formatos de datos.

5.4.1 JSON

JSON (Javascript Object Notation) [40] es un formato de texto ligero para el intercambio de datos. Es independiente de cualquier lenguaje, pero utiliza convenciones ampliamente conocidas por los programadores. Concretamente está constituido por dos tipos de estructuras, una colección de pares clave-valor y una lista ordenada de valores. Estas dos estructuras mencionadas son universales, dado que todos los lenguajes las soportan de una forma u otra.

5.4.2 XML

XML (Extensible Markup Language) [41] es un meta-lenguaje que permite definir lenguajes de marcado. Estos lenguajes se caracterizan por ser extensibles, estructurados y estar orientados al contenido y no a la presentación. Una de sus utilidades consiste en representar información estructurada, de modo que puede ser almacenada, transmitida y procesada por diversos tipos de dispositivos y aplicaciones.

5.5 Herramientas auxiliares

5.5.1 REM

REM (Requirements Management) [42] es una herramienta de gestión de requisitos diseñada para soportar la fase de ingeniería de requisitos de un proyecto software. La herramienta proporciona ayuda para generar la documentación de un proyecto software. Así mismo permite documentar los actores, objetivos y requisitos, tanto funcionales como no funcionales de un proyecto software.

5.5.2 Visual Paradigm

Visual Paradigm [43] es una herramienta CASE diseñada para soportar el ciclo de vida completo del proceso de desarrollo de un producto software. Esta herramienta permite realizar una gran cantidad de diagramas distintos con lo que llevar a cabo una representación completa del producto software.

Entre las características más destacables de esta herramienta se pueden mencionar:

1. Disponible para múltiples plataformas.
2. Licencia gratuita.
3. Diseño centrado en casos de uso.
4. Uso de un lenguaje estándar común a todo el equipo de desarrollo para facilitar la comunicación.

5.5.3 Sublime Text 3

Sublime Text 3 [\[44\]](#) es un editor de código multiplataforma y ligero que se caracteriza porque permite resaltar la sintaxis de múltiples lenguajes. Dispone además de autoguardado, numerosas opciones de personalización, así como edición de código y automatización de tareas.

Capítulo 6. Conclusiones y trabajo futuro

Con respecto a los objetivos planteados al inicio del proyecto, decir que se han alcanzado y cumplido todos con éxito. Se ha desarrollado un servicio de valor añadido capaz de explotar la información que proporciona el sensor de geolocalización que poseen los Smartphone.

Este servicio es capaz de recopilar información de elementos de interés cultural de la ciudad de Madrid haciendo uso del repositorio de datos abiertos que proporciona el ayuntamiento de Madrid, y del que se puede recuperar información sobre museos, monumentos, parques, iglesias u oficinas de turismo entre otros.

Por último, decir que se ha conseguido desarrollar una aplicación móvil Android que satisface los requisitos establecidos inicialmente. La aplicación permite mostrar los elementos de interés cultural de Madrid que se encuentran dentro del radio de proximidad considerado por el usuario. De igual modo, la aplicación es capaz de mostrar rutas hasta los elementos de interés cultural utilizando el transporte público gracias a utilizar también el repositorio de datos abiertos de la EMT de Madrid.

Este proyecto puede considerarse una primera aproximación sobre cómo puede explotarse el concepto IOT para crear servicios de valor añadido en el ámbito de las visitas culturales a una ciudad. Sin embargo, se puede profundizar en el uso de la IOT. A continuación, se plantean algunas líneas de trabajo futuro.

En este trabajo solo se hace uso de la geolocalización del Smartphone para recuperar los elementos de interés cultural cercanos al usuario. Sin embargo, no se están utilizando el resto de los sensores que están disponibles en un dispositivo móvil. Así por ejemplo, usando otros sensores se podría conocer la afluencia de personas que tiene cada día de la semana, o el tiempo medio de visita de los usuarios a los elementos de interés cultural.

Otra posible aplicación de los sensores sería la comunicación con un Wearable tal como un reloj inteligente para conocer el número medio de pasos que se dan en un museo o monumento con el objetivo de saber si las visitas se realizan de forma exhaustiva o no.

Una importante mejora consistiría en el despliegue y desarrollo de una infraestructura de sensores en cada elemento de interés cultural de manera que cada uno de ellos tuviera sus propios sensores de temperatura, humedad o presencia. A partir de la información obtenida de estos sensores, se podría crear en tiempo real rutas turísticas en las que los usuarios pudiesen priorizar qué elementos de interés cultural visitar primero y cuales después. Por ejemplo, visitar primero aquellos en los que haya menos visitantes, o bien organizar una ruta para visitar museos o monumentos al aire libre en horarios que haga menos calor o frío según la estación del año.

Por último, otra línea de trabajo futuro sería ampliar el tipo de elementos de interés cultural gestionados por la aplicación. En este trabajo se han tratado con museos, monumentos, parques, iglesias y oficinas de turismo. Sin embargo, se podría recuperar

información sobre bibliotecas, instalaciones deportivas, eventos que puedan tener lugar durante periodos cortos de tiempo, ferias, embajadas, consulados y muchos otros más.

Chapter 6. Conclusions

Regarding the objectives set at the beginning of the project, it has to be said that all have been successfully achieved. An added value service has been developed capable of exploiting the information provided by the geolocation sensor that the Smartphone possesses.

This service is capable of collecting information on elements of cultural interest of the city of Madrid using the open data repository provided by the city council of Madrid, and from which information about museums, monuments, parks, churches or tourist offices can be retrieved among others.

Finally, say that it has managed to develop an Android mobile application that meets the requirements established initially. The application allows to show the elements of cultural interest of Madrid that are within the proximity radius considered by the user. In the same way, the application is able to show routes to the elements of cultural interest using public transport thanks to also use the open data repository of the EMT in Madrid.

This project can be considered a first approach on how the IOT concept can be exploited to create value-added services in the field of cultural visits to a city. However, the use of IOT can be deepened. Here are some lines of future work.

In this work, only the geolocation of the Smartphone is used to recover the elements of cultural interest close to the user. However, the rest of the sensors that are available on a mobile device are not being used. Thus, for example, using other sensors, one could know the influx of people they have each day of the week, or the average visit time of the users to the elements of cultural interest.

Another possible application of the sensors would be the communication with a Wearable such as a smart watch to know the average number of steps taken in a museum or monument in order to know if the visits are carried out exhaustively or not.

An important improvement would be the deployment and development of a sensor infrastructure in each element of cultural interest so that each of them had its own temperature, humidity or presence sensors. From the information obtained from these sensors, tourist routes could be created in real time in which users could prioritize which elements of cultural interest to visit first and which later. For example, first to visit those in which there are fewer visitors or organize a route to visit museums or outdoor monuments at times that are less hot or cold depending on the season of the year.

Finally, another line of future work would be to expand the type of cultural interest elements managed by the application. In this work they have dealt with museums, monuments, parks, churches and tourist offices. However, information about libraries, sports facilities, events that may take place during short periods of time, fairs, embassies, consulates and many others could be retrieved.

Bibliografía

- [1] Cristoffer Adrián. IoT: origen, importancia en el presente y perspectiva de futuro. Recuperado de <https://www.itop.es/blog/item/iot-origen-importancia-en-el-presente-y-perspectiva-de-futuro.html> . 2018
- [2] ¿Qué es el valor añadido de una empresa? Recuperado de <https://simpleproductividad.es/blog/valor-anadido-influencia-empresa/>
- [3] ¿Qué es un sensor? Recuperado de <https://profesormolina1.webcindario.com/tecnologia/sens-transduct/que-es.htm>
- [4] 12 sensores que encontrarás en tu móvil: cómo funcionan y para qué sirven. Recuperado de <https://www.xataka.com/basics/12-sensores-que-encontraras-tu-movil-sirven>
- [5] Giroscopio en el móvil: estos son todos los usos menos conocidos. Recuperado de <https://www.xatakandroid.com/aplicaciones-android/giroscopio-en-el-movil-estos-son-todos-los-usos-menos-conocidos>
- [6] Los 9 sensores más importantes de tu móvil. Recuperado de <https://andro4all.com/2018/12/sensores-movil>
- [7] ¿Cómo funciona un sensor biométrico? Recuperado de <https://blog.gruponovelec.com/redes-vdi/como-funciona-sensor-biometrico/>
- [8] ¿Qué son los datos abiertos? Recuperado de <http://governobert.gencat.cat/es/dades-obertes/que-son-les-dades-obertes/>
- [9] Datos abiertos y reutilización de la información de la información del sector público. Recuperado de https://administracion.gob.es/pag_Home/espanaAdmon/Transparencia-y-datos-abiertos-2/datos-abiertos.html#.XWBJnC0rxQI
- [10] ¿Qué son los datos abiertos? Recuperado de <http://opendatahandbook.org/guide/es/what-is-open-data/>
- [11] Los cinco problemas técnicos más frecuentes a la hora de publicar tus datos abiertos. Recuperado de <https://datos.gob.es/es/noticia/los-cinco-problemas-tecnicos-mas-frecuentes-la-hora-de-publicar-tus-datos-abiertos>
- [12] Datos de investigación: Repositorios de datos abiertos. Recuperado de <https://guiasbuh.uhu.es/c.php?g=498100&p=3907565>

- [13] Proyecto Aporta. Recuperado de [https://es.wikipedia.org/wiki/Proyecto_Aporta#Catálogo de datos abiertos](https://es.wikipedia.org/wiki/Proyecto_Aporta#Catálogo_de_datos_abiertos)
- [14] ¿Qué contiene el portal? Recuperado de <https://datos.madrid.es/portal/site/egob/menuitem.400a817358ce98c34e937436a8a409a0/?vgnextoid=23ed21f248bf1610VgnVCM1000001d4a900aRCRD&vgnnextchannel=23ed21f248bf1610VgnVCM1000001d4a900aRCRD&vgnnextfmt=default>
- [15] Geolocalización. Recuperado de <https://es.wikipedia.org/wiki/Geolocalización>
- [16] Tecnologías y sistemas de localización: Una oportunidad para las pymes. Recuperado de <https://ticnegocios.camaravalencia.com/servicios/tendencias/tecnologias-de-localizacion-una-oportunidad-para-las-pymes/>
- [17] ¿Qué es la geolocalización y cómo funciona? Recuperado de <https://www.marketinhouse.es/que-es-la-geolocalizacion/>
- [18] Wi-Fi Location. Qué es, como funciona y para qué sirve este estándar de geoposicionamiento en interiores con Wi-Fi. Recuperado de <https://www.redeszone.net/2017/03/11/wi-fi-location-funciona-sirve-este-estandar-geoposicionamiento-interiores-wi-fi/>
- [19] Geosocialización: Geolocalización y redes sociales. Recuperado de <https://www.goodrebels.com/es/la-geosocializacion-geolocalizacion-y-redes-sociales-ii/>
- [20] Pokemon Go. Recuperado de: https://es.wikipedia.org/wiki/Pokémon_GO
- [21] Madrid's Best: City Travel Guide Recuperado de: https://play.google.com/store/apps/details?id=com.touchscreentravels.madridsbest&hl=es_419
- [22] Mapa de Madrid offline. Recuperado de: https://play.google.com/store/apps/details?id=org.mapapps.mapyourtown.madrid&hl=es_419
- [23] Google Maps. Recuperado de: https://play.google.com/store/apps/details?id=com.google.android.apps.maps&hl=es_419
- [24] Foursquare: Qué es, como funciona y para qué sirve. Reccuperado de <https://elblogdeeva.com/2014/05/19/foursquare-para-empresas-y-community-managers/>
- [25] ¿Qué es Yelp y en qué se diferencia de otras plataformas? Recuperado de <https://victormartinp.com/que-es-yelp-y-en-que-se-diferencia-de-otras-plataformas/>

- [26] Python. Recuperado de <https://es.wikipedia.org/wiki/Python>
- [27] ¿Qué es y para qué sirve Python? Recuperado de <https://noticias.universia.es/ciencia-tecnologia/noticia/2017/07/19/1154393/sirve-phyton.html>
- [28] ¿Qué es Python? Recuperado de <https://desarrolloweb.com/articulos/1325.php>
- [29] ¿Qué es la tecnología Java y para qué la necesito? Recuperado de https://www.java.com/es/download/faq/whatis_java.xml
- [30] Conceptos básicos del lenguaje Java. Recuperado de <https://www.ibm.com/developerworks/ssa/java/tutorials/j-introtojava1/index.html>
- [31] ¿Qué es PHP? Recuperado de <https://www.php.net/manual/es/intro-whatism.php>
- [32] PHPMailer potente y sencilla clase para el envío de email desde PHP. Recuperado de <https://desarrolloweb.com/articulos/phpmailer.html>
- [33] Realizar peticiones HTTP con la librería Volley en Android. <http://www.hermosaprogramacion.com/2015/02/android-volley-peticiones-http/>
- [34] Android. Recuperado de <https://es.wikipedia.org/wiki/Android>
- [35] Introducción a Android Studio. Recuperado de <https://developer.android.com/studio/intro?hl=es-419>
- [36] ¿Qué es la computación en la nube? Recuperado de <http://www.ictea.com/cs/index.php?rp=/knowledgebase/215/What-is-Cloud-Computing.html>
- [37] IaaS, PaaS y SaaS. Modelos de servicio de IBM Cloud. Recuperado de <https://www.ibm.com/es-es/cloud/learn/iaas-paas-saas>
- [38] ¿Qué es Google Cloud Platform? Recuperado de <https://www.doctormetrics.com/google-cloud-platform/>
- [39] Google Cloud Functions Tutorial: What is Google Cloud Functions? Recuperado de <https://medium.com/@iromin/google-cloud-functions-tutorial-what-is-google-cloud-functions-8796fa07fc7a>
- [40] Introducción a JSON. Recuperado de <https://www.json.org/json-es.html>
- [41] XML ¿Qué es? Recuperado de <http://www.mundolinux.info/que-es-xml.htm>

[42] Herramienta REM. Recuperado de http://www.lsi.us.es/descargas/descarga_programas.php?id=3

[43] Visual Paradigm. Recuperado de https://www.ecured.cu/Visual_Paradigm

[44] Sublime Text, un sofisticado editor de código multiplataforma. Recuperado de <https://www.genbeta.com/herramientas/sublime-text-un-sofisticado-editor-de-codigo-multiplataforma>

[45] Scrum en la metodología ágil: te explicamos que es y como funciona. Recuperado de <https://clouding.io/blog/scrum-en-la-metodologia-agil-te-explicamos-que-es-y-como-funciona/>

[46] Arquitectura Cliente Servidor. Recuperado de https://www.ecured.cu/Arquitectura_Cliente_Servidor

[47] EMT de Madrid. Recuperado de https://twitter.com/EMTmadrid?ref_src=twsrc%5Egoogle%7Ctwcamp%5Eserp%7Ctwgr%5Eauthor

[48] Concepto de Google Maps. Recuperado de <https://neoattack.com/neowiki/google-maps/>

Apéndice 1 - Manual de usuario

A1.1 Introducción

En este apéndice se describe un manual de usuario de la aplicación desarrollada. Para ello se van a mostrar las distintas pantallas que forman parte de la aplicación y se explica la función que realiza cada uno de los elementos que forman parte de ellas.

A1.2 Pantalla de inicio

La pantalla de inicio es la primera que se muestra cuando el usuario inicia la aplicación (Figura 1).

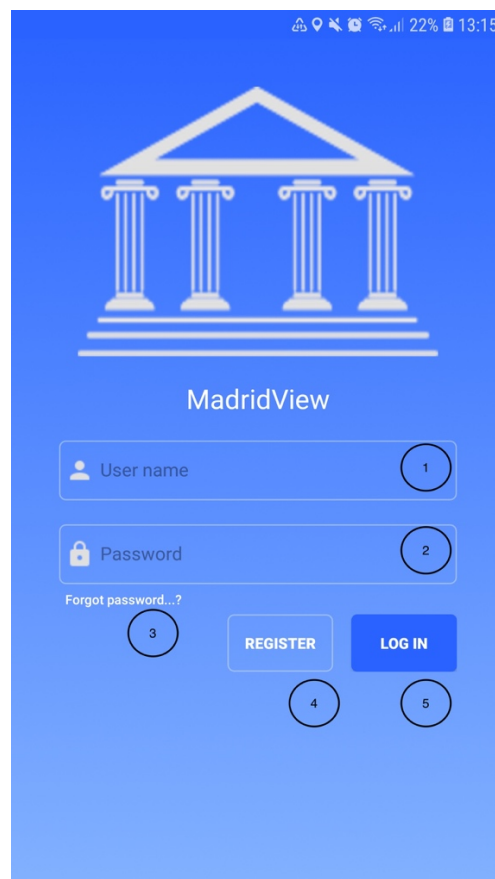


Figura 1. Pantalla de Inicio

Los elementos que se pueden observar en la Figura 1 son los siguientes:

1. Cuadro de texto en el que el usuario debe de introducir el nombre de usuario con el que previamente se ha registrado.
2. Cuadro de texto en el que el usuario debe introducir la contraseña con la que se ha registrado o con una elegida previamente por el usuario.

3. Botón que muestra al usuario una nueva pantalla en la que se indican los pasos a seguir para establecer una nueva contraseña.
4. Botón que muestra una nueva pantalla en la que se le indica al usuario los pasos a realizar para registrarse en el sistema.
5. Botón que el usuario debe presionar para iniciar sesión en el sistema. Previamente el usuario debe de introducir tanto su nombre de usuario como la contraseña. En caso de que la información sea correcta, el usuario iniciará sesión en el sistema, y en caso contrario se le mostrará al usuario la razón por la que no ha podido iniciar sesión.

A1.3 Pantalla de registro

La pantalla de registro (Figura 3) es aquella pantalla en la que se mostrará un formulario donde el usuario deberá de introducir la información necesaria para registrarse en el sistema.

Figura 2. Pantalla de registro

Los elementos que contiene la pantalla de registro son los siguientes:

1. Cuadro de texto en el que el usuario debe introducir el nombre de usuario con el que identificarse en el sistema. La única restricción sobre el nombre de usuario es que debe ser único en el sistema.
2. Cuadro de texto en el que el usuario debe introducir una dirección de correo válida que sirva de contacto con el usuario.
3. Cuadro de texto en el que el usuario debe de introducir la contraseña para iniciar sesión en el sistema.
4. Cuadro de texto en el que el usuario debe introducir la misma contraseña que el cuadro de texto anterior para asegurarse de que la contraseña con la que se va a registrar en el sistema es la que desea por el usuario.
5. Botón que debe presionar el usuario para registrarse y validar la información introducida. En caso de que haya algún error se le notificará al usuario cual es el error. En caso de que la información sea correcta se iniciará sesión en el sistema, y se enviará un correo (Figura 3.a) a la dirección de correo introducida dando la bienvenida al usuario. Además a través del correo enviado, el usuario podrá verificar su cuenta usando el enlace que aparece en el mismo (Figura 3.b).

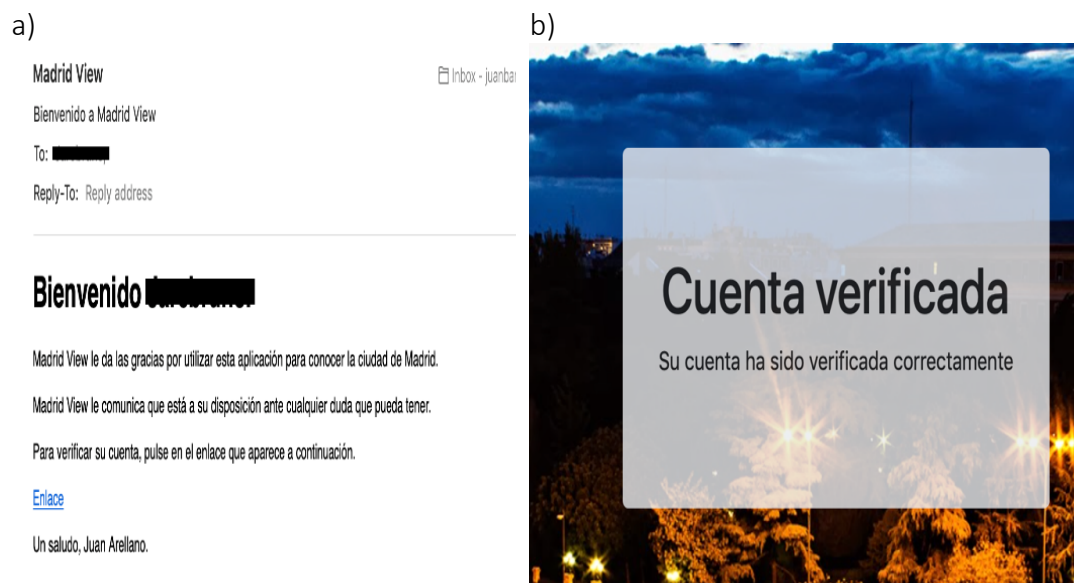


Figura 3. a) Correo de bienvenida. b) Mensaje de confirmación.

A1.4 Pantalla de recuperación de contraseña

En esta pantalla (Figura 4) se le mostrará un formulario al usuario que debe rellenar para poder recuperar la contraseña.

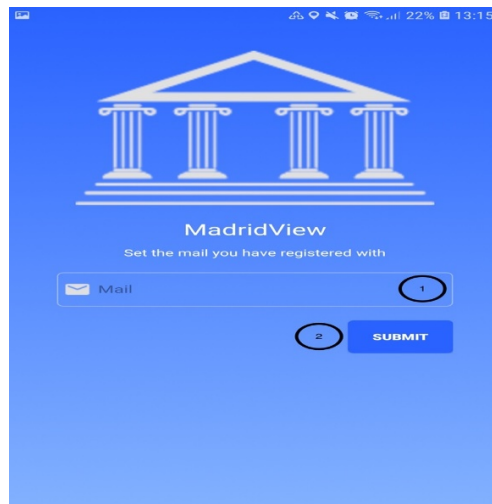


Figura 4. Pantalla de recuperación de contraseña

Los elementos que se pueden apreciar en dicha pantalla son:

1. Cuadro de texto en el que el usuario debe de introducir la dirección de correo con la que previamente se registró.
2. Botón que el usuario debe presionar para que el sistema reestablezca la contraseña. En caso de que exista algún error se le notificará al usuario, y en caso contrario se reestablecerá la contraseña del usuario notificando la nueva en la dirección de correo introducida.

A1.5 Pantalla de inicio

Una vez que el usuario inicia sesión accede a la pantalla de inicio (Figura 6.a) desde la que puede hacer uso de la funcionalidad de la aplicación.

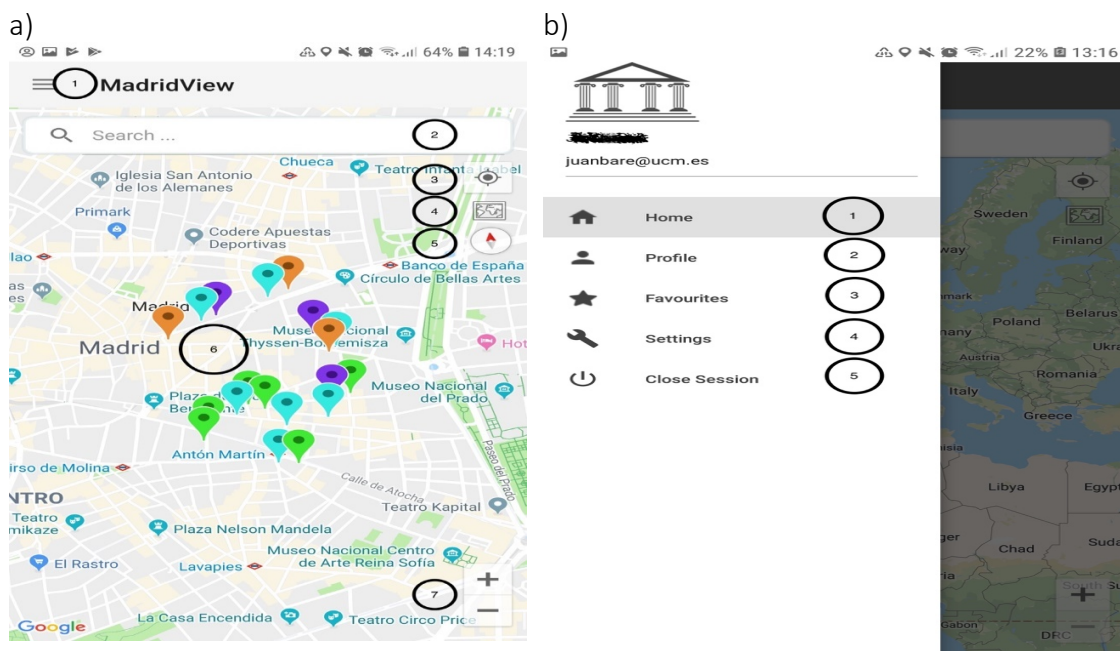


Figura 4. a) Pantalla de Inicio, b) Menú lateral de la aplicación

Las funciones de la pantalla de inicio son las siguientes:

1. En la Figura 6.b se muestra el menú lateral de la pantalla de inicio con las distintas opciones que puede usar el usuario para interactuar con la aplicación:
 - 1.1. Botón que muestra la pantalla de inicio de la aplicación.
 - 1.2. Botón que muestra la pantalla desde la que el usuario puede gestionar su propia información.
 - 1.3. Botón que muestra la pantalla en la que aparecen los elementos culturales favoritos marcados previamente por el usuario desde la pantalla de inicio de la aplicación.
 - 1.4. Botón que muestra los ajustes de la aplicación.
 - 1.5. Botón que permite al usuario cerrar la sesión que tiene iniciada en la aplicación
2. Barra búsqueda en la que el usuario es capaz de introducir el nombre un elemento cultural específico para que se muestre en el mapa.
3. Botón que permite centrar el mapa sobre la ubicación del usuario, la cual se mostrará como un punto azul.
4. Botón que permite modificar el estilo del mapa que se está mostrando. Tras pulsarlo, se mostrará en el centro de la pantalla un cuadro de dialogo desde el que el usuario podrá seleccionar la opción que más le agrade (Figura 7).

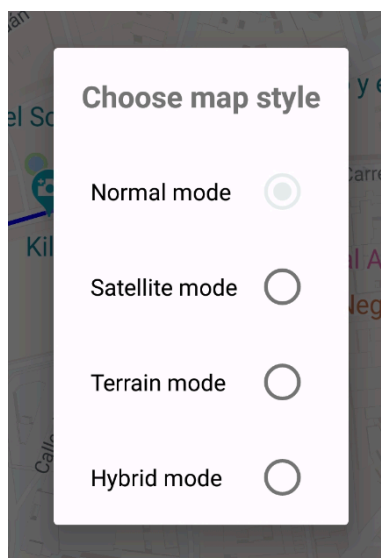


Figura 7. Opciones de estilos para el mapa

5. Botón que permite centrar el mapa entorno a los puntos cardinales Norte, Sur, Este y Oeste.

6. Elementos de interés cultural. Se muestran en el mapa en forma de marcador. El color de los marcadores varía en función del tipo que sean. Los tipos de elementos que se recogen en el sistema son:

6.1. Museo.

6.2. Monumento.

6.3. Parque.

6.4. Iglesia católica.

6.5. Oficina de turismo.

7. Botones que permiten ajustar el zoom sobre el mapa

Cuando el usuario presiona un marcador del mapa, aparece un cuadro de dialogo flotante con la información referente al elemento de interés cultural asociado a dicho marcador.



Figura 8. Información de un elemento de interés cultural

En el cuadro de dialogo (Figura 8) se puede apreciar la información que el sistema conoce sobre dicho elemento de interés cultural tal como una descripción de este, la dirección en la que se encuentra, el horario de este, y el número de personas que acuden a este elemento de interés cultural durante cada uno de los días de la semana. Además, en dicho cuadro de dialogo, aparecen los siguientes elementos:

1. Imagen del elemento de interés cultural.
2. Nombre del elemento de interés cultural.
3. Tipo del elemento de interés cultural.
4. Botón que permite al usuario marcar el elemento de interés cultural como favorito.
5. Botón que permite al usuario buscar la ruta hasta el elemento de interés cultural desde el punto en el que se encuentra. Una vez se presiona este botón, en la pestaña de inicio aparece una ventana flotante (Figura 9) con un pequeño resumen de la ruta hasta dicho elemento de interés cultural.

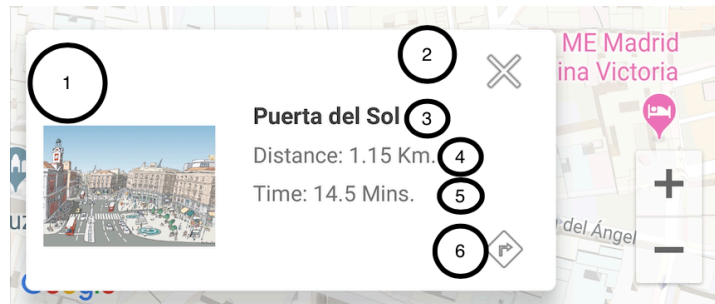


Figura 9. Resumen de la ruta

Los elementos que forman parte de dicha ventana flotante son:

1. Imagen del elemento de interés cultural.
2. Botón que permite cancelar la ruta hasta el elemento de interés cultural.
3. Nombre del elemento de interés cultural.
4. Distancia desde el punto en el que se encontraba el usuario cuando se quiso conocer la ruta hasta el elemento de interés cultural.
5. Tiempo desde el punto en el que se encontraba el usuario cuando se quiso conocer la ruta hasta el elemento de interés cultural.
6. Botón que muestra un cuadro de dialogo (Figura 10) en el centro de la pantalla con los detalles de la ruta hasta el elemento de interés cultural.

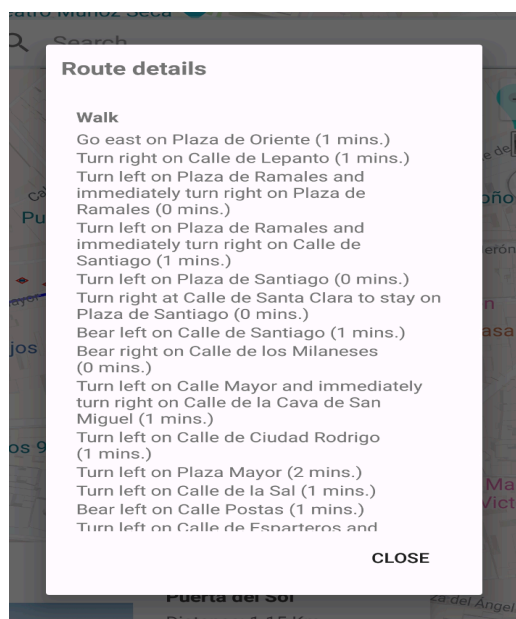


Figura 10. Itinerario de la ruta

A1.6 Pantalla del perfil del usuario

La pantalla del perfil del usuario (Figura 11) es aquella pantalla que permite realizar al usuario la gestión de su propia información, así como la posibilidad de deshabilitar su cuenta o cambiar la contraseña con la que accede al sistema.

The screenshot shows a mobile application interface for a user profile. At the top, there's a header with a hamburger menu icon and the title 'Profile'. Below this, a user profile section displays a silhouette icon, the email 'juanbare@ucm.es', and a 'Verified' status with a green checkmark. A circled '1' is next to the email. Below the profile section are two main functional areas. The first is 'Change Password', which includes a subtext: 'For security reasons, you need to set your old password and write twice your new password'. It contains three input fields: 'Set old password' (with a lock icon and a circled '2'), 'Set password' (with a lock icon and a circled '3'), and 'Repeat password' (with a lock icon and a circled '4'). A blue 'CHANGE PASSWORD' button (with a circled '5') is at the bottom of this section. The second area is 'Deactivate Account', with subtext: 'If you deactivate your account, it will be removed in 30 days if you do not login again'. It features a single input field labeled 'Password' (with a lock icon and a circled '6'). A red 'DEACTIVATE ACCOUNT' button (with a circled '7') is at the bottom of this section. The top status bar shows various icons and '21%' battery.

Figura 11 Pantalla del perfil del usuario

La pantalla está compuesta de los siguientes elementos:

1. Información del usuario tal como el nombre de usuario, la dirección de correo con la que se registró el usuario, o información sobre si la cuenta del usuario está o no verificada.
2. Cuadro de texto donde el usuario deberá escribir la contraseña que tiene activa en caso de que quiera establecer una nueva contraseña.
3. Cuadro de texto donde el usuario deberá escribir la nueva contraseña con la que la que va a poder acceder al sistema.
4. Cuadro de texto donde el usuario deberá confirmar la nueva contraseña que desea establecer.
5. Botón que debe pulsar el usuario para realizar el cambio de contraseña. En caso de que haya algún error se le comunicará al usuario, y en caso contrario la contraseña del usuario se verá cambiada.
6. Cuadro de texto en el que el usuario deberá escribir la contraseña que tiene activa en caso de que quiera deshabilitar su cuenta.
7. Botón que debe pulsar el usuario una vez escrita la contraseña en el cuadro de texto anterior para confirmar que desea deshabilitar su cuenta.

A1.7 Pantalla de favoritos

La pantalla de favoritos (Figura 12) es aquella en la que se listan todos los elementos de interés cultural marcados por el usuario desde la pantalla de inicio de la aplicación.

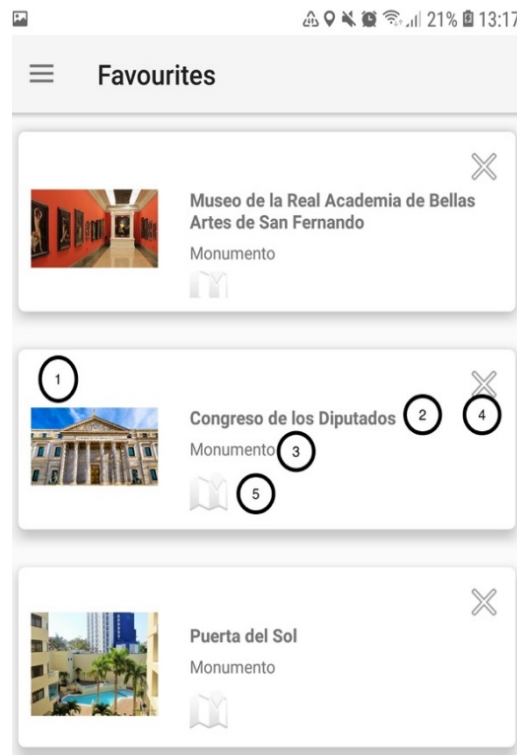


Figura 12. Pantalla de favoritos

Cada uno de los elementos que forman el listado de favoritos está compuesto de los siguientes elementos:

1. Imagen del elemento de interés cultural marcado como favorito.
2. Nombre del elemento de interés cultural marcado como favorito.
3. Tipo del elemento de interés cultural marcado como favorito.
4. Botón que permite al usuario eliminar de la lista de favoritos un elemento de interés cultural.
5. Botón que permite al usuario mostrar el elemento de interés cultural en la pantalla de inicio.

En caso de que el usuario no haya marcado ningún elemento como favorito, esta lista estará vacía.

A1.8 Pantalla de ajustes

Desde la pantalla de ajustes (Figura 13) el usuario puede realizar cambios en la configuración de la aplicación tales como fijar el tipo de elementos de interés cultural

buscados, el color de los marcadores que aparecen en el mapa, o la distancia donde se buscarán los elementos de interés cultural.

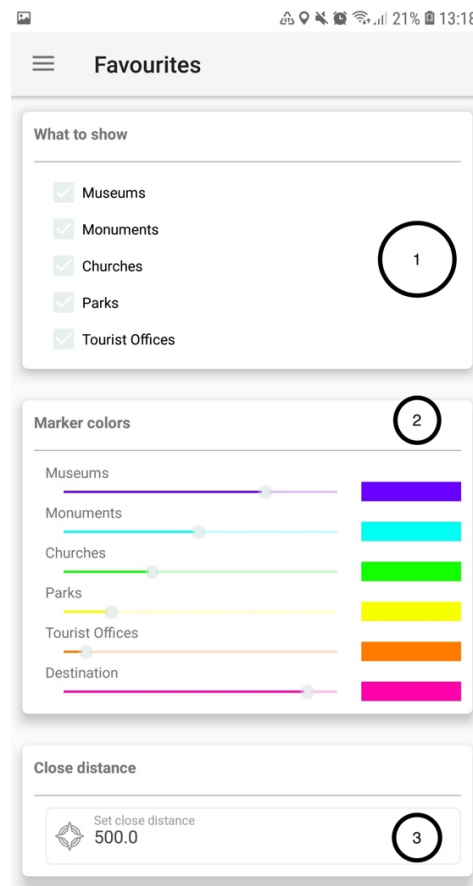


Figura 13. Pantalla de ajustes

Los elementos que forman parte de la pantalla de la pantalla de ajustes son:

1. Sección en la que el usuario establece los elementos que desea mostrar en el mapa de la pantalla de inicio.
2. Sección en la que el usuario establece el color de los marcadores que se muestran en el mapa de la pantalla de inicio.
3. Cuadro de texto donde el usuario indica la distancia en la que desea buscar elementos de interés cultural.